



**AALBORG UNIVERSITY**  
DENMARK



**SINTEF**

DEPARTMENT OF CIVIL ENGINEERING

MASTER THESIS IN CIVIL & STRUCTURAL ENGINEERING

---

***Implementation and comparison of two numerical models for trawl cod-ends***

---

*By Mihhail Samusev*

Supervised by  
Thomas Lykke Andersen (AAU)  
Bent Herrmann (SINTEF OCEAN)

**June 8, 2018**





**AALBORG UNIVERSITET**  
STUDENTERRAPPORT

**MSc. Structural & Civil Engineering**  
Thomas Manns vej 23  
9220 Aalborg East

**Title:**

Implementation and comparison of two numerical models for trawl cod-ends

**Theme:**

Numerical simulation of fishing gear, finite element modelling, object-oriented programming

**Project period:**

September 1<sup>st</sup> 2017  
to June 8<sup>th</sup> 2018

**Supervisor(s):**

Thomas Lykke Andersen (AAU)  
Bent Herrmann (SINTEF OCEAN)

**Number of pages:**

52

**Number of pages in appendices:**

28

**Abstract:**

A cod-end is the rearmost part of a trawl fishing gear that collects the catch during the towing. The shape of cod-ends is of importance as it determines mesh opening and consequently influences the selectivity of fish from the cod-end. The project considers two different numerical models of the cod-end deformation: the axis-symmetric model and the 3D finite element model based on the triangular elements. The models are described and implemented in the C# code using the principles of object-oriented programming. Different solutions for improving the convergence of the Newton-Raphson method that is commonly used to find the equilibrium shape of the cod-end models are presented and compared. The numerical results from the two models are compared. A flume tank experiment is performed with a motion tracking system to sample the cod-end shape in 3D. The results suggests a good correspondence between the models for the small scale cod-ends with thin twines.

**Participant(s):**

---

Mihhail Samusev

## **Preface**

This thesis has been submitted to fulfill the study curriculum of 3<sup>rd</sup> and 4<sup>nd</sup> semester of the M.Sc. programme in Structural and Civil Engineering at Aalborg University. The project work has been carried out in the period between the 1<sup>st</sup> of September 2017 to the 8<sup>th</sup> of June 2018 in cooperation with SINTEF OCEAN Hirtshals Denmark. The thesis is equivalent to 60 ECTS.

The project has been supervised by Thomas Lykke Andersen from the AAU side and Bent Herrmann from the SINTEF OCEAN side. I would like to thank the supervisors for the cooperation. Furthermore, i would like to thank Kurt Hansen, Anders Nielsen and Morten Jensen from SINTEF OCEAN for their help with the experiment at flume tank in Hirtshals Denmark.

---

---

# Contents

---

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Background . . . . .	1
1.1.1	Cod-end . . . . .	1
1.1.2	Netting characteristics . . . . .	2
1.1.3	Challenges related to selectivity . . . . .	3
1.2	State of the art in cod-end shape estimation . . . . .	4
1.3	Thesis purpose and limitations . . . . .	5
1.4	Outline of the thesis report . . . . .	5
<b>2</b>	<b>Mathematical models</b>	<b>7</b>
2.1	3D Model based on triangular elements . . . . .	7
2.1.1	Coordinate systems . . . . .	7
2.1.2	Twine vectors . . . . .	8
2.1.3	Force models . . . . .	9
2.2	Axis-symmetric cod-end models . . . . .	16
2.2.1	T0 model . . . . .	17
2.2.2	T90 model . . . . .	20
<b>3</b>	<b>Computer implementation and solution algorithm</b>	<b>23</b>
3.1	Program structure and classes' description . . . . .	23
3.1.1	3D model . . . . .	23
3.1.2	Axis-symmetric model . . . . .	25
3.1.3	Input-output . . . . .	25
3.2	Meshing and assembly of 3D cod-end model . . . . .	25
3.3	Choice of mathematical library . . . . .	26
3.4	Solution to non-linear system . . . . .	28
<b>4</b>	<b>Model comparison</b>	<b>35</b>
4.1	Mesh convergence study . . . . .	35
4.2	Comparison 1 - basic test . . . . .	36
4.3	Comparison 2 - Effect of the knot size . . . . .	36
4.4	Comparison 3 - effect of the drag on the netting . . . . .	37
4.4.1	Netting drag applied to all twines . . . . .	39
4.4.2	Netting drag applied to twines not in contact with the catch . . . . .	40
4.5	Comparison 4 - effect of mesh opening . . . . .	41
4.6	Discussion . . . . .	42
<b>5</b>	<b>Validation</b>	<b>43</b>
5.1	Shape estimation of a single panel cod-end . . . . .	43
5.1.1	Setup . . . . .	43
5.1.2	Results . . . . .	45
5.1.3	Conclusion . . . . .	47
<b>6</b>	<b>Conclusion and recommendations for future work</b>	<b>49</b>

---

CONTENTS	1
<b>Bibliography</b>	<b>52</b>
<b>A Derivatives for 3d FEM model</b>	<b>iii</b>
<b>B Derivatives for the axis-symmetric models</b>	<b>ix</b>
<b>C Line search algorithm</b>	<b>xi</b>
<b>D Numerical tests on the axis-symmetric model</b>	<b>xiii</b>
<b>E Mesh refinement and smoothing</b>	<b>xxv</b>
<b>F SINTEF OCEAN flume tank in Hirtshals</b>	<b>xxix</b>

---

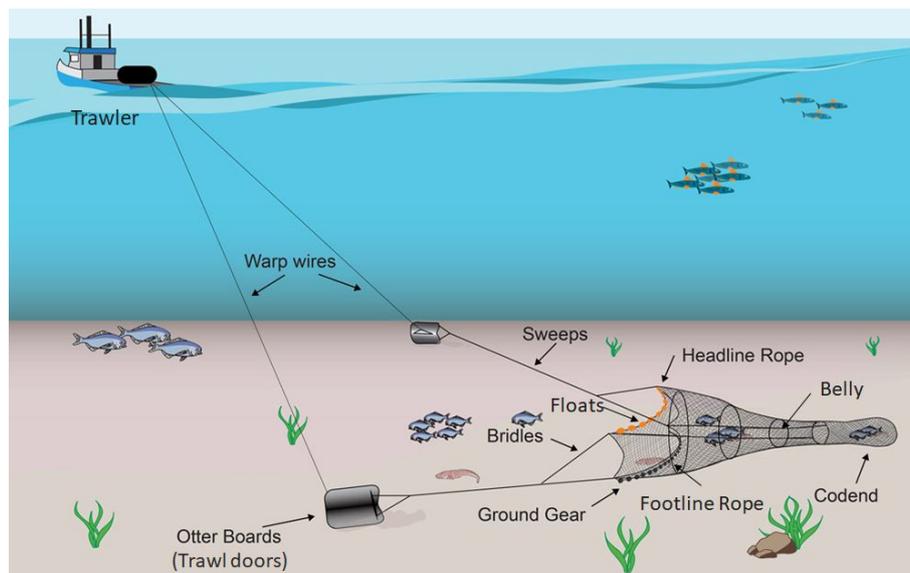


## CHAPTER 1

# Introduction

## 1.1 Background

Trawling is a method of fishing that involves dragging a cone or a funnel shaped fishing net (trawl) through the water column behind one or two boats (trawlers). Trawls are divided into two main types depending on whether they are towed along the sea floor or in mid water. A bottom (demersal) trawl is a trawl which is towed along or close to the seafloor and used to catch demersal fish species such as cod, haddock, sand eel and shrimp. In contrast to the bottom trawling, mid-water (pelagic) trawling is characterized by the fishing gear not coming into contact with the seabed during fishing and used to fish pelagic species for example herring, mackerel, blue whiting, tuna and sardines. One of the differences between a bottom trawl and a mid-water trawl is that the latter does not require a ground gear to facilitate the tow across often coarse sea bottom. Typical components of a bottom trawl are presented in Figure 1.1, p. 1.



**Figure 1.1:** Key components of a bottom trawl [1].

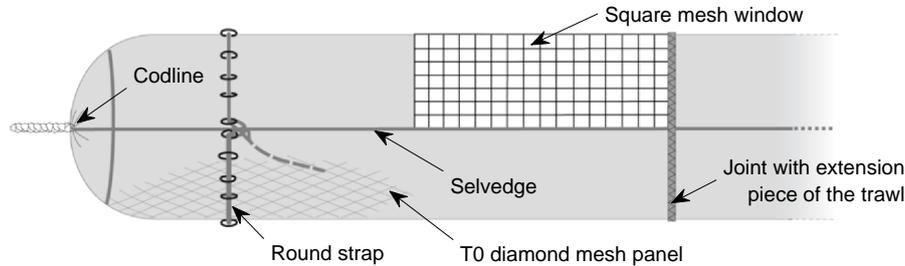
The main goals pursued by a trawl designer is to maximize the catch of the target species with the largest size, at the same time maintaining a low amount of by-catch, which is defined as juvenile fish or other marine species that are caught unintentionally.

Catch maximization is provided by opening the trawl in the vertical and horizontal direction. Vertical opening is maintained by the correct combination of floats on the headline rope and weights or ground gear on the footline rope. Horizontal opening is maintained by using the otter boards (trawl doors) that diverge from each other during towing due to their geometry. The extent of the horizontal opening is also influenced by the towing speed.

### 1.1.1 Cod-end

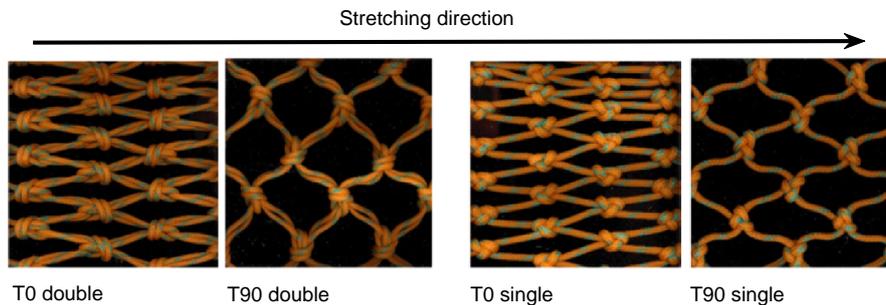
Trapped inside the trawl, the catch is led through the trawl belly and the extension piece towards the cod-end where the catch is accumulated. One end of the cod-end is joined to the rest of the trawl through an extension piece, while the other is tightened by a rope called cod-line fed through the last row of meshes. A typical

cod-end is often made of two or four rectangular panels of netting joined at the sides by selvages. Selvages are created by gathering few edge meshes of joint panels and binding them together. Often cod-end panels are made of diamond mesh netting, square mesh netting or the combination of both as shown in Figure 1.2, p. 2. Evaluation of a cod-end selectivity is a complex task. It depends on many parameters and their interaction, such as towing speed, current amount of catch, material and geometry of the meshes, additional elements attached to the cod-end structure, etc.



**Figure 1.2:** Bacoma cod-end with a round strap for lifting the catch aboard. Square mesh window takes half of the top panel [2].

The choice of the mesh type highly influences the selectivity of the cod-end. For example a diamond mesh can be either conventional (T0) or 90° turned (T90) as shown in Figure 1.3, p. 2. In the traditional T0 netting orientation of the standard cod-end, the mesh bending resistance of the twines tends to close the meshes. This mechanical behaviour is called mesh resistance to opening. Turning the netting 90° reverses this mechanism, which provides a more open mesh [3]. The choice of the best netting type is not a trivial task, since different types of meshing might improve the selectivity for say round fish species, but worsen it for the square fish species.



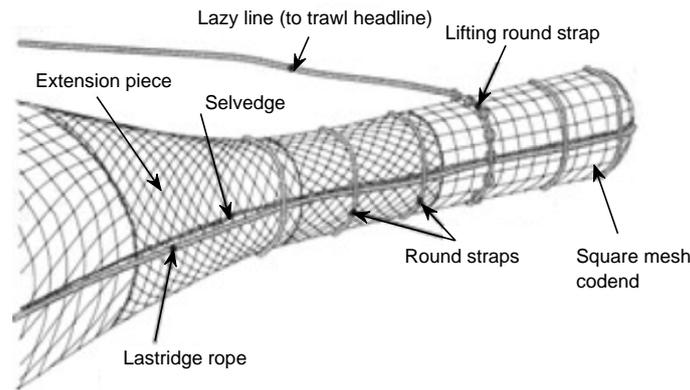
**Figure 1.3:** Stretching of T0 and T90 diamond meshes made of a single and a double twine.

Bigger cod-ends that are expected to carry large amount of catch are strengthened by attaching ropes lengthwise to prevent stretching or circumstantially to restrict the diameter under the pressure of accumulated fish [4]. These ropes are called lastridge ropes and round straps respectively, but also can be referred as strengthening ropes (See Figure 1.4, p. 3).

The strengthening ropes affect the mesh openness and thus the selectivity of the cod-end. Lastridge ropes are shorter than the stretched length of the netting to which they are attached. The ropes bear the tension that would usually be in the mesh twines allowing the meshes to open and deform more easily. The extent of this effect will depend on the length ratios of the rope and the netting [5]. Round straps are needed to divide the catch in a cod-end into parts of a size convenient for lifting and handling aboard the vessel. The presence of the straps limits the mesh openness in the circumference of the cod-end in the proximity of the strap. Apart from that, in some seas the cod-ends are supplied with sorting devices such as sea lion and turtle exclusion devices, that are out of the scope of this report.

### 1.1.2 Netting characteristics

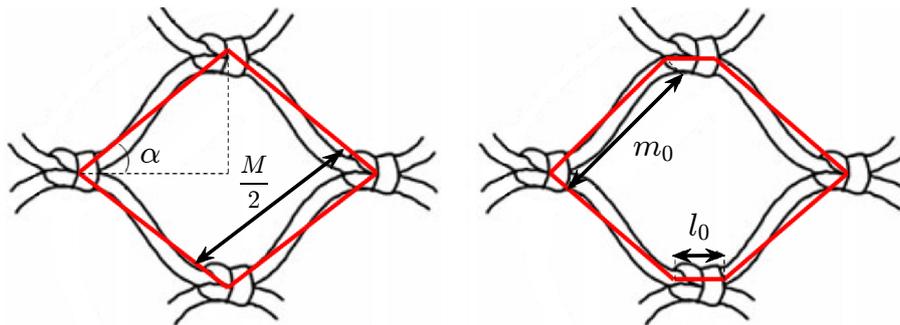
Netting is made of the textile material such as nylon, polyester and polypropylene which differs from other structural material by having a unique combination of flexibility, anisotropy and uneven surface structure.



**Figure 1.4:** Square mesh cod-end with various strengthening ropes [2].

The material anisotropy results in the behaviour where netting can only effectively resist tensile loading at the same time having low or in some cases negligible resistance in compression, bending and torsion [6].

A basic unit of a netting panel is a mesh which is defined by four twines and four knots as shown in Figure 1.5, p. 3. The geometry of the mesh is often described through a mesh side  $M$  which is double of the unstretched twine length  $m_0$ . In the diamond mesh representation the twine length can be measured between the knot centers. However, for the meshes with bigger twine diameter  $D$ , the knot size  $l_0$  becomes significant, and the hexagonal mesh representation is used. The opening angle  $\alpha$  is measured as a half angle between the twines emerging from the same knot.



**Figure 1.5:** Geometric parameters of a mesh. The diamond representation (left) and the hexagonal representation (right) of the mesh.

### 1.1.3 Challenges related to selectivity

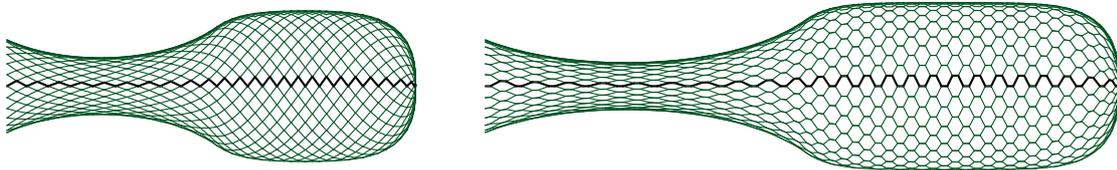
Mesh openness during trawling have a major effect on fish sizes accumulating in the cod-end. Therefore, knowing the mesh geometry in trawl cod-ends during trawling is of importance for being able to select efficient cod-end designs for different fisheries. Reliable estimation of the cod-end shape during trawling is directly connected to improvement the catch selectivity, thus reducing the negative environmental impact of trawling. Previously, studies of trawl selectivity have been carried out in full scale at the sea. However, due to the large number of uncontrollable parameters, numerous trials have to be undertaken in order to obtain a reasonable amount of data for statistical analysis. This leads to expensive studies, that have not proven to be effective. It is mostly due to the fact that one cannot extrapolate the experimental data to predict the selectivity due to new combinations of parameters outside the scope of the experiment [7][8].

To overcome this limitation, with the development of numerical methods it was possible to use predictive computer models of cod-end selectivity. The models such as PRESEMO [9], have been developed during the last decade and are able to simulate cod-end selectivity with respect to different cod-end parameters. Even though these tools are based on approximations, their results are often reliable. However, it is important to know the fish behaviour and the mechanical cod-end behaviour to have an accurate prediction.

## 1.2 State of the art in cod-end shape estimation

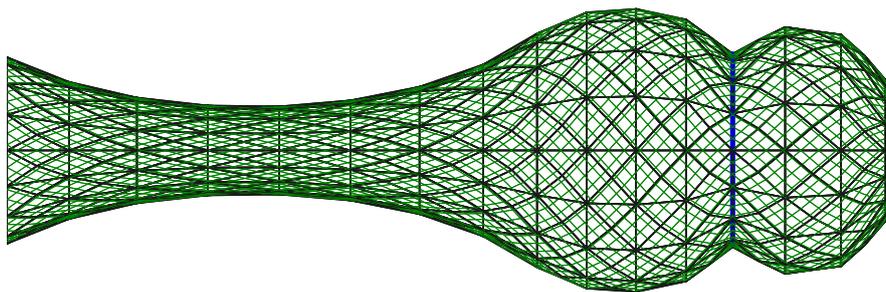
In general, the cod-end can be regarded as a very flexible structure with very large potential deformations due to its geometry and the netting material behaviour. In this respect, the modelling of the cod-end shape is more complicated than other structures with only small deformations (building components, machine parts) as it involves geometric non-linearity.

A cod-end that consists of only one type of meshes can be reasonably considered axis-symmetric. Under this assumption O'Neill [10] have developed an analytical model where he derived differential equations that govern the geometry of axis-symmetric cod-ends for a range of different mesh shapes. Priour et al. [8] have developed a numerical axis-symmetric model based on finite element method by looking at the force balance on the twine elements on one row along the cod-end length (Highlighted in Figure 1.6, p. 4). The model is valid for the cod-end made up of diamond and rectangular meshes. However, when the twines become thicker, the knot size affects the mesh geometry, that rather resembles a hexagon than a diamond as was shown in Figure 1.5, p. 3. It was found that the hexagonal mesh is more suitable to describe the actual shape of the meshes in the cod-ends. Therefore, Priour [7] has extended the previous axis-symmetric model to account for knot size by the hexagonal mesh description.



**Figure 1.6:** Axis-symmetric models with 25 meshes along, 25 meshes around and 13 meshes blocked by catch. The diamond mesh representation (left) and the hexagonal mesh representation (right) with  $l_0 = 0.5m_0$ .

In the case of the axis-symmetric model, only the forces symmetric about the cod-end axis can be considered. Those forces are the twine tensions and the hydrodynamic forces. As a downside of the model, the asymmetric forces such as catch weight, buoyancy forces cannot be accounted for. The operational conditions when towing happens with an angle to the cod-end axis can neither be simulated. However, the biggest limitation is that only possible to model cod-ends consisting of one type of meshes, and no additional attachments. This fact disallows to test a realistic cod-end with different strengthening ropes as shown in Figure 1.4, p. 3. Furthermore, the numerical axis-symmetric models rely on the assumption that twines have a negligible bending stiffness and are fully compressible.



**Figure 1.7:** A 3D cod-end model based on triangular finite elements with a round strap.

A more general 3D FEM model based on triangular elements is developed by Priour [11]. Priour formulates a triangular element for netting where the boundaries are not necessary parallel to the twines or to the mesh diagonals. Each vertex is linked to the net, so when the equilibrium position of each vertex is found, the equilibrium position of the net is found. The model takes into account the twine tension, the drag force on the net due to the current, the pressure created by the catch in the cod-end, the buoyancy and weight of the net, the mesh opening stiffness and the bending stiffness of the net. This FEM model resulted in non-commercial finite element package FEMNET developed at IFREMER. Comparisons between previously mentioned analytic axis-symmetric model and 3D FEM model with triangular elements have been made [12] and showed a relatively good results. A big advantage of triangular elements is that they are able to cover a

large number of meshes (Figure 1.7, p. 4), in contrast to the 3D models, where netting is discretized with bar or beam elements for each twine as suggested by Tsukrov et al. [13]. Due to the high number of degrees of freedom the time required for the determination of the equilibrium position for the bar based models is generally long. In this sense both numerical axis-symmetric and 3d model with triangular elements are more computationally efficient.

In recent years, there is a trend towards the use of netting materials manufactured with thicker and stiffer twines. The increased mesh resistance to opening of such materials as well as compression resistance especially at the knots has a notable impact on the cod-end deformation. The problem was investigated by O'Neill [14], Morvan et al. [15], Prada and González [16] and Sala et al. [17], that resulted in the development of several analytical, numerical and experimental approaches for non-linear stiffness models of a net twine. Those more advanced twine models, however, were never combined with the numerical axis-symmetric models. The analytical axis-symmetric model by O'Neill on the other hand considers this mechanical behaviour, and shows the results that are in good correspondence [12] with a more general numerical 3D model for the netting deformation.

The equilibrium shapes of netting structures modeled with FEM are normally calculated by solving the equilibrium equations of the model with the iterative Newton–Raphson (NR) method [11]. In contrary, de la Prada and González [18] have assessed suitability of gradient-based energy minimization methods to calculate the equilibrium shape of netting structures. Their tests on triangular FEM models indicate that the L-BFGS (Limited memory Broyden–Fletcher–Goldfarb–Shanno) has a great potential to replace or complement the Newton–Raphson method in the equilibrium analysis. Nevertheless, similarly to NR, L-BFGS suffers from convergence problems when the initial shape is bad. Generation of a good initial shape for the numerical cod-end model has not been well documented.

Some of the aforementioned models were validated by flume tank testing [11]. In the tank a tested cod-end with a catch is subjected to current and its shape contour is captured by a digital camera. the contour obtained experimentally is then compared to the numerically obtained contour of the cod-end. As an improvement, it is possible to use motion capturing systems to get a spatial representation of the cod-end shape. Madsen et al. [3] used QUALISYS [19] for a qualitative comparison of different cod-end designs. QUALISYS finds its application in such fields as bio-mechanics, robotics and film industry. The system uses a stereo principle with a number of underwater cameras, that by means of triangulation can detect the position of reflective markers being placed on the object of interest. Applied to the field of fishing gear design this approach has a potential of allowing more quantitative validation of the numerical models.

### 1.3 Thesis purpose and limitations

The purpose of this master thesis is to document an implementation and comparison of two different numerical models for estimation of cod-end shape and towing resistance during towing. Main milestones of the thesis are:

1. Describe the derivation and features of the two numerical models: axis-symmetric model and 3D FEM model based on planar triangular elements.
2. Implement the models in a computer program using a general purpose programming language C# and Object Oriented Programming paradigm. Describe classes and their responsibilities, as well as other critical implementation details. As a limitation, user interface is not developed.
3. Allow the numerical models to be applied independently, but also explore how the models can be combined to increase the performance. Research and apply the methods to obtain a sufficient speed and robustness of the calculation routine.
4. Compare the numerical models in order to identify when the faster axis-symmetric model can be used instead of the slower, yet more detailed 3D model and yield the same results.
5. Validate the models against a flume tank test in a quantitative way by comparing the numerical models to a test cod-end shape sampled with a 3D motion tracking system.

### 1.4 Outline of the thesis report

The rest of the thesis is organized as follows:

---

- Chapter 2, p. 7 presents mathematical-physical models of the axis-symmetric and the triangular finite element models.
  - Chapter 3, p. 23 presents the computer implementation and the solution algorithms.
  - Chapter 4, p. 35 presents the numerical comparison between the two models.
  - Chapter 5, p. 43 presents the validation of the two models with the experiment from the flume tank.
  - Chapter 6, p. 49 presents the conclusion and discussion of the future work.
-

---

# Mathematical models

---

In this chapter two numerical cod-end models are described. The models are: the 3D FEM model based on the triangular elements [11] and the axis-symmetric model with hexagonal mesh representation [7]. The goal of this chapter is to present the underlying model assumptions and derivation of the force equations, that are used later to find the equilibrium of the structure.

Since the problem at hand is a flexible netting structure, the forces in both models are geometrically non-linear. The equilibrium shape of the structure is achieved when a deformed configuration of the structure  $\mathbf{X}$  provides the force equilibrium to the system  $\mathbf{F}(\mathbf{X}) = \mathbf{0}$ . Commonly for netting structures, the resulting equation system is solved iteratively with Newton-Raphson method [20].

$$\mathbf{J}(\mathbf{X}_i) \mathbf{h}_i = -\mathbf{F}(\mathbf{X}_i) \quad (2.1)$$

$$\mathbf{X}_{i+1} = \mathbf{X}_i + \mathbf{h}_i \quad (2.2)$$

Here the non-linear system is linearized with Taylor approximation and solved for displacement  $\mathbf{h}$ . The deformed configuration of the structure is updated by the displacement to yield a new configuration for the linearization. The process is repeated until the force equilibrium is achieved. The classic version of the method relies on the analytical representation of force Jacobian  $\mathbf{J}(\mathbf{X})$  that is intentionally omitted in the original papers describing the models. Here the force derivatives forming Jacobian matrices are included in Appendix B and Appendix A.

## 2.1 3D Model based on triangular elements

In this section the 3D FEM model for cod-ends is considered. The model is based on the planar triangular elements developed by Priour [11]. Triangular elements offer high flexibility in both geometric modelling and numerical calculation aspects.

From the geometric modelling perspective, the triangles are widely used to approximate the curvature of an arbitrary 3D shape. There are many algorithms available for mesh refinement and smoothing, some of which are applied to the problem at hand Section 3.4, p. 30. In the particular finite element method, a triangular element works as a super element that combined all the effects of the twines contained within it. This allows to avoid the problems typical for the models where individual physical twines are modeled by bar elements [13] also known as numerical twines. For those models the bars must be aligned to the physical twines of the netting which limits the designer's freedom of joining netting panels made of different meshes.

From the numerical calculation perspective, the triangular element allows to reduce the total amount of d.o.f. in the model and thus the calculation time compared to the models based on bar elements. Due to the fact that position of each vertex of a triangle depends on the forces acting on the netting, the equilibrium of the entire net is found when the equilibrium of all the vertices is found.

### 2.1.1 Coordinate systems

Apart from the standard Cartesian coordinates, the 3D FEM model operates in two additional panel-related 2D coordinate systems that are presented in Figure 2.1, p. 8.

Mesh coordinates  $u_i$  and  $v_i$  indicate how many meshes there are between the origin and the target point  $i$  along two Cartesian vectors  $\mathbf{u}$  and  $\mathbf{v}$ . The  $\mathbf{u}$  and  $\mathbf{v}$  direction vectors vary from triangle to triangle and depend on the global netting deformation. The deformation, however, does not change the mesh coordinates of the nodes. For example, node 3 in Figure 2.1, p. 8 (b) is located 4 meshes along  $\mathbf{u}$  direction and 3 meshes along  $\mathbf{v}$  direction from the origin independent on the panel deformation. The mesh coordinates are used by net

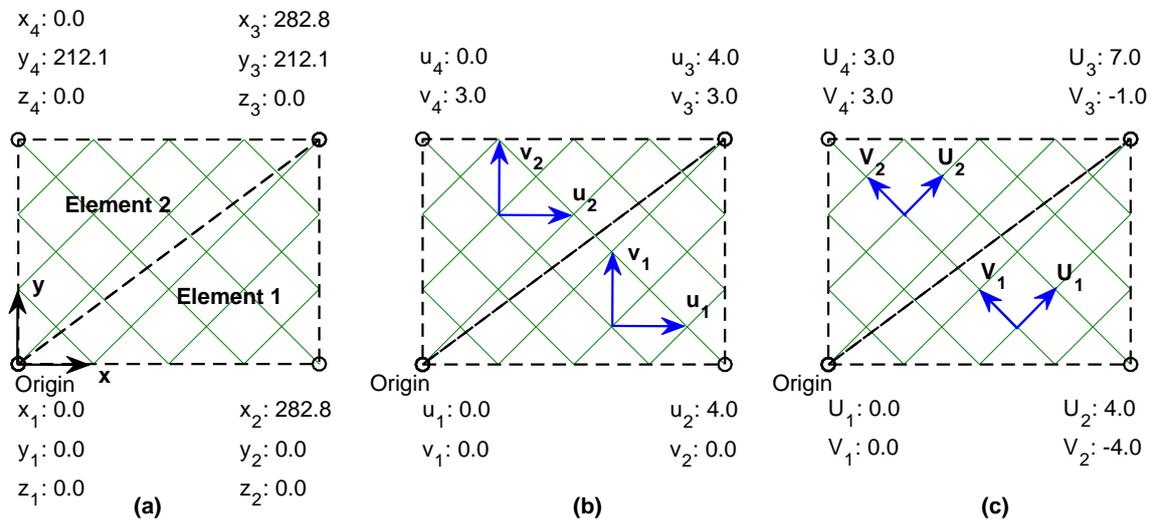
---

designers and net makers to describe the width and length of a netting panel and also to position additional elements on the panel, such as cuts, strengthening ropes, rings etc.

Twine coordinates  $U_i$  and  $V_i$  follow the same principle. They indicate how many twines there are between the origin and the target point  $i$  along two Cartesian vectors  $\mathbf{U}$  and  $\mathbf{V}$ . Similarly to the mesh coordinates the twine coordinates of the vertices are constant under the netting deformation. The  $\mathbf{U}$  and  $\mathbf{V}$  directions vary from triangle to triangle and depend on the netting deformation. For example, node 3 in Figure 2.1, p. 8 (c) is located 7 twines along  $\mathbf{U}$  direction and 1 twine along negative  $\mathbf{V}$  direction from the origin. Twine coordinates are used in the numerical model to calculate the Cartesian coordinates and the lengths of the twine vectors  $\mathbf{U}$  and  $\mathbf{V}$ . This information allows to define the nodal forces due to twine elongation or contraction. Furthermore, it gives twine orientation relative to each other to determine their opening, as well as to the towing direction for the application of drag forces. Since the input is usually given in terms of the mesh coordinates (design coordinates), the conversion between the coordinates as shown in Eq. (2.3).

$$U_i = u_i + v_i \quad (2.3)$$

$$V_i = v_i - u_i$$



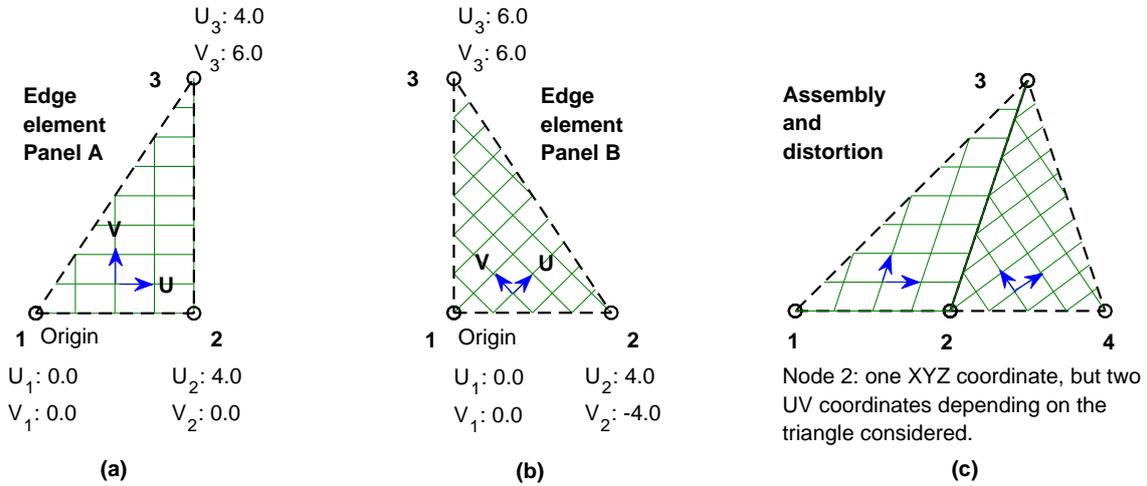
**Figure 2.1:** Underformed 4 by 3 meshes panel with mesh side of 100 mm and initial opening angle of  $45^\circ$  is presented in 3 coordinate systems: **a** - Cartesian coordinates, **b** - mesh coordinates, **c** - twine coordinates.

The Figure 2.2, p. 9 presents how two different netting panels are joined. The joint between two panels is created by merging the vertices that lay on the common edge. Displacement of the vertex 3 has an effect on the netting, which can be represented as a change in length and the direction of the  $\mathbf{U}$  and  $\mathbf{V}$  vectors. As seen, the physical twines do not have to meet on the common edge. It should be noted, that both mesh and twine coordinates are unique for each triangle. In a special case as for a panel made of the same meshes (for example Figure 2.1, p. 8) if a pair of triangles share the same vertex, they obtain a duplicate set of mesh and twine coordinates. On the other hand, if the pair of triangles share the same vertex that lies on the edge between two panels, each triangle will have a unique set of mesh and twine coordinates for the same vertex.

Remarkably, this formulation allows for the twine coordinates  $U$  and  $V$  to take any decimal number, therefore, it is not required that the nodes of the triangle coincide with the physical knots of the netting or the triangle contains the integer amount of meshes.

### 2.1.2 Twine vectors

Since deformation of the netting inside the triangle can be described by  $\mathbf{U}$  and  $\mathbf{V}$  twine vectors, the forces and the tangent stiffness matrices in the method depend on the relative lengths and orientations of these twine vectors. The vectors in turn depend on the current deformation state in the Cartesian coordinates, as well as the description of the netting in the twine coordinates. Applying the assumption that the twines remain



**Figure 2.2:** Joining two panels with different meshes: **a** - square mesh panel, **b** - diamond mesh panel, **c** - panels are assembled and distorted, nodes are re-ordered.

parallel under deformation, the edges of a triangle can be represented as a linear combination of  $\mathbf{U}$  and  $\mathbf{V}$  vectors as shown Eq. (2.4).

$$\mathbf{S}_{12} = \begin{pmatrix} x_2 - x_1 \\ y_2 - y_1 \\ z_2 - z_1 \end{pmatrix} = (U_2 - U_1)\mathbf{U} + (V_2 - V_1)\mathbf{V} \quad (2.4)$$

$$\mathbf{S}_{13} = \begin{pmatrix} x_3 - x_1 \\ y_3 - y_1 \\ z_3 - z_1 \end{pmatrix} = (U_3 - U_1)\mathbf{U} + (V_3 - V_1)\mathbf{V}$$

Both Cartesian coordinates and twine coordinates are known for each triangle, therefore the  $\mathbf{U}$  and  $\mathbf{V}$  are defined as shown in Eq. (2.5). The coefficient  $d$  has a physical meaning. It is defined as a cross product between triangle's edges  $\mathbf{S}_{12}$  and  $\mathbf{S}_{13}$  expressed in the twine coordinates. Therefore,  $d$  equals double the amount of the twines in a single direction and also the double amount of knots. For example, considering Figure 2.2, p. 9 (b)  $d$  is calculated to be 48, therefore there are 24 twines in both  $\mathbf{U}$  and  $\mathbf{V}$  directions and 24 knots.

$$\mathbf{U} = \frac{V_3 - V_1}{d}\mathbf{S}_{12} - \frac{V_2 - V_1}{d}\mathbf{S}_{13} \quad (2.5)$$

$$\mathbf{V} = \frac{U_2 - U_1}{d}\mathbf{S}_{13} - \frac{U_3 - U_1}{d}\mathbf{S}_{12}$$

where:

$$d = (U_2 - U_1)(V_3 - V_1) - (U_3 - U_1)(V_2 - V_1)$$

### 2.1.3 Force models

The forces on the model can be categorized into two groups, internal and external forces. The internal forces are the conservative forces due to the elastic deformation of the netting and the following resistance of the twines to tension, opening and bending. The external forces are due to drag on the netting, catch pressure and the weight of the cod-end. The total forces are the sum of all the external and internal forces.

$$\mathbf{F}^{total} = \mathbf{F}^{internal} + \mathbf{F}^{external} = \mathbf{F}^{tension} + \mathbf{F}^{open} + \mathbf{F}^{bend} + \mathbf{F}^{weight} + \mathbf{F}^{drag} + \mathbf{F}^{catch} \quad (2.6)$$

#### Twine tension

It is assumed that in each triangular element the twines along the same direction have equal and constant strains. Thus, all twines along  $\mathbf{U}$  direction have the same length  $|\mathbf{U}|$  and all twines along  $\mathbf{V}$  direction have the

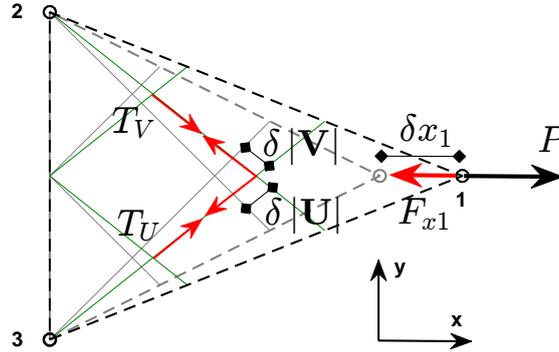
same length  $|\mathbf{V}|$ . Tensile force in a twine is calculated based on its length relative to the unstretched length  $m_0$  as shown in Eq. (2.8). The twines behave anisotropically, meaning their resistance in compression is significantly lower than in tension. This effect is taken into account by applying the appropriate axial rigidity  $EA$  depending on whether the twine is stretched or compressed. Factor  $k$  in the formula is smaller than 1 and typically equal or less than 1 % [18].

$$\begin{aligned} T_U &= EA \frac{|\mathbf{U}| - m_0}{m_0} \\ T_V &= EA \frac{|\mathbf{V}| - m_0}{m_0} \end{aligned} \quad (2.7)$$

where:

$$EA = \begin{cases} EA_{tension} & \text{if } |\mathbf{U}| > m_0 \\ k EA_{tension} & \text{if } |\mathbf{U}| \leq m_0 \end{cases} \quad (2.8)$$

The transformation from twine tensions inside the element to nodal forces is established through the Principle of Virtual Work. As exemplified in Figure 2.3, p. 10, the external work is produced as the force  $P$  initiates the displacement of node 1 by the amount  $\delta x_1$  along the positive  $x$ -direction. Consequently, the twines are deformed by the amounts  $\delta|\mathbf{U}|$  and  $\delta|\mathbf{V}|$ , which produces the internal work of the tensile forces  $T_U$  and  $T_V$ . The internal work is negative, because the force and the displacement are in the opposite directions. There are  $\frac{d}{2}$  twines of each kind that contribute to the internal work. Equilibrium demands that the total work done by the system is zero as shown in Eq. (2.9).



**Figure 2.3:** Principle of Virtual Work for the force  $F_{x1}$  applied to node 1 in positive  $x$ -direction.

$$\delta W = \delta W_{int} + \delta W_{ext} = P \delta x_1 - (T_U \delta|\mathbf{U}| + T_V \delta|\mathbf{V}|) \frac{d}{2} = 0 \quad (2.9)$$

Diving both sides of the equation by the infinitesimal displacement  $\delta x_1$  and letting it approach zero the expression for the nodal force  $F_{x1}$  is obtained in Eq. (2.10). Minus sign in the equation appears due to the equilibrium condition on the node  $P + F_{x1} = 0$ .

$$F_{x1} = - \left( T_U \frac{\partial |\mathbf{U}|}{\partial x_1} + T_V \frac{\partial |\mathbf{V}|}{\partial x_1} \right) \frac{d}{2} \quad (2.10)$$

Applying the Eq. (2.10) to all nodes in all three directions yields the complete set of nodal forces due to twine tension for the triangular element Eq. (2.11).

$$\begin{aligned}\mathbf{F}_1 &= (V_3 - V_2)T_U \frac{\mathbf{U}}{2|\mathbf{U}|} + (U_2 - U_3)T_V \frac{\mathbf{V}}{2|\mathbf{V}|} \\ \mathbf{F}_2 &= (V_1 - V_3)T_U \frac{\mathbf{U}}{2|\mathbf{U}|} + (U_3 - U_1)T_V \frac{\mathbf{V}}{2|\mathbf{V}|} \\ \mathbf{F}_3 &= (V_2 - V_1)T_U \frac{\mathbf{U}}{2|\mathbf{U}|} + (U_1 - U_2)T_V \frac{\mathbf{V}}{2|\mathbf{V}|}\end{aligned}\quad (2.11)$$

The derivatives of the twine lengths as well as the Jacobian matrix of the force vector used for the Newton-Raphson method are found in Appendix A.

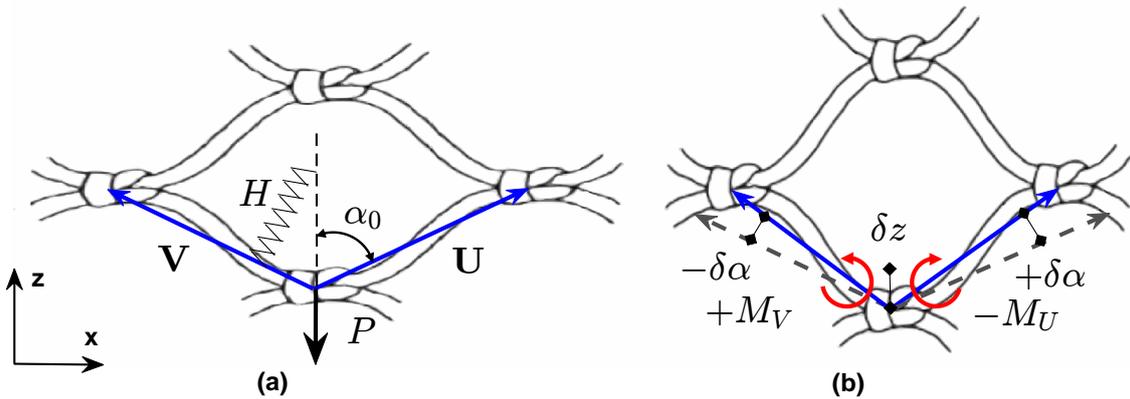
### Resistance to opening

Mesh panels made of thin twines resist in plane deformation mostly in tension. However, if the thicker twines are used, bending resistance becomes significant. The bending resistance that happens in-plane of the netting panel is also known as mesh resistance to opening. So far, the most detailed opening resistance models were developed and validated for the net structures composed of bar elements [16] [15]. For the triangular element [11] proposes a simplified mesh opening resistance model. Key assumption is that the angle between the twines is in linear relationship with the force couple  $M$  created by the twines on the knot. The opening angle  $\alpha$  is defined as the half angle between the twines in Eq. (2.12). Equal and opposite force couples  $M_U$  and  $M_V$  occur when the  $\mathbf{U}$  and  $\mathbf{V}$  twines form a half angle that is different from an initial opening angle  $\alpha_0$  as shown in Eq. (2.13). The proportionality coefficient between the force couple and the angle is mesh opening stiffness  $H$ . Interaction between tension and bending (tension stiffening) is not considered in the model.

$$\alpha = \frac{1}{2} \arccos \left( \frac{\mathbf{U} \cdot \mathbf{V}}{|\mathbf{U}| |\mathbf{V}|} \right) \quad (2.12)$$

$$M = H(\alpha - \alpha_0) \quad (2.13)$$

The transformation from force couples inside the element into nodal forces is established through the Principle of Virtual Work. As exemplified in Figure 2.4, p. 11, the external work is produced as the force  $P$  initiates the displacement of a knot by the amount  $\delta z$  along the negative  $z$ -direction. Consequently the angle  $\alpha_0$  becomes smaller by an amount  $\delta\alpha$ , producing force couples  $M_U$  and  $M_V$  on the knot and thus the external work. The counterclockwise rotation is positive in the sign convention for the right hand side Cartesian coordinates. There are  $\frac{d}{2}$  knots that contribute to the internal work. Equilibrium demands that the total work done is zero as shown in Eq. (2.14).



**Figure 2.4:** A mesh with opening stiffness  $H$  and initial opening angle  $\alpha_0$ . The force  $P$  stretches the mesh along  $z$ -direction (mesh works as a T90 mesh), initiating the moments  $M_U$  and  $M_V$  to resist the deformation.

$$\delta W = \delta W_{int} + \delta W_{ext} = P\delta z + (-M_U \delta\alpha + M_V (-\delta\alpha)) \frac{d}{2} = P\delta z - Md\delta\alpha \quad (2.14)$$

Diving both sides of the equation by infinitesimal displacement  $\delta z$  and letting it approach zero the expression for the external force  $P$ . In order to express the resisting force  $F_z$  in Eq. (2.15) minus sign appears due to the equilibrium condition on the node  $P + F_z = 0$ .

$$F_z = -H(\alpha - \alpha_0)d \frac{\partial \alpha}{\partial z} \quad (2.15)$$

Applying the Eq. (2.15) to all nodes in all three directions yields the complete vector of 9 nodal forces due to the mesh opening resistance Eq. (2.16).

$$\mathbf{F}^{open} = -H(\alpha - \alpha_0)d \frac{\partial \alpha}{\partial \mathbf{k}} \quad (2.16)$$

where:

$$\mathbf{k} = (x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3)$$

The expressions for the gradient of the opening angle and the Jacobian of the force vector used for the Newton-Raphson method are found in Appendix A.

### Resistance to bending

When thicker twines are used, netting panels not only have a higher in-plane bending resistance, but also a higher out of plane bending resistance. The effect of out-of-plane bending is taken into account with a simplified approach similar to the in-plane bending. In the current model, bending happens if two triangles that share a common edge are not co-planar. In this case, there are non-zero angles  $\alpha$  and  $\beta$  between the  $\mathbf{U}$  twines and  $\mathbf{V}$  twines of the two triangles ( $a$  and  $b$ ) as shown in Eq. (2.17). The angles are constant along the common edge. The key assumption is that the bending moment along the common edge is proportional to the bending stiffness  $EI$  and the curvature of radius  $R$  as shown in Eq. (2.18). Interaction between tension and bending (tension stiffening) is also not considered in the model.

$$\alpha = \arccos\left(\frac{\mathbf{U}_a \cdot \mathbf{U}_b}{|\mathbf{U}_a| |\mathbf{U}_b|}\right) \quad \beta = \arccos\left(\frac{\mathbf{V}_a \cdot \mathbf{V}_b}{|\mathbf{V}_a| |\mathbf{V}_b|}\right) \quad (2.17)$$

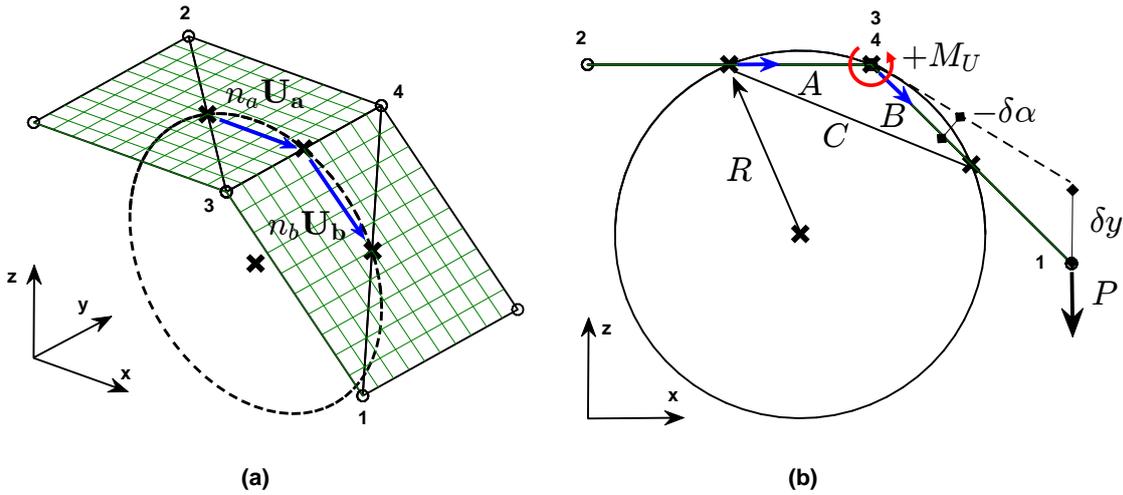
$$M_U = \frac{EI}{R_U} \quad M_V = \frac{EI}{R_V} \quad (2.18)$$

In order to estimate the radius of the curvature  $R$ , a circle is fit to three points that lay on the pair of the bent triangles: the point on the common edge, and a point on each of the triangles that is an average number of twines away from the common edge (See Figure 2.5, p. 13 a). The points form an inscribed triangle with sides  $A$ ,  $B$  and  $C$  expressed in Eq. (2.19). Notably, Figure 2.5, p. 13 shows a special situation when only  $U$ -twines are bent.  $V$ -twines remain parallel and do not contribute to bending in this case. In general case, there are two unique fitting circles to estimate the curvature radiuses for both  $U$  and  $V$ -twines.

$$\begin{aligned} A &= |n_a \mathbf{U}_a| \\ B &= |n_b \mathbf{U}_b| \\ C &= |n_a \mathbf{U}_a + n_b \mathbf{U}_b| \end{aligned} \quad (2.19)$$

Where the average numbers of  $U$ -twines  $n_a$  and  $n_b$  for triangles  $a$  and  $b$  are calculated depending on the number of  $U$ -twines of the triangle  $\frac{d}{2}$ , and the amount of twine rows that bend over the common edge (See Eq. (2.20)).

$$\begin{aligned} n_a &= \left\lfloor \frac{d_a}{2V_{edge}} \right\rfloor = \left\lfloor \frac{d_a}{2(V_3 - V_4)} \right\rfloor \\ n_b &= \left\lfloor \frac{d_b}{2V_{edge}} \right\rfloor = \left\lfloor \frac{d_b}{2(V_3 - V_4)} \right\rfloor \end{aligned} \quad (2.20)$$



**Figure 2.5:** A square panel is discretized into four triangles and bent in the middle. Because of the angle between the  $U$ -twines of the two triangles  $a$  and  $b$ , they are no longer co-planar and the bending moment  $M_U$  appears to resist this deformation. Under the deformation all the  $V$ -twines remain parallel, therefore they do not contribute to the bending resistance.

The curvature radius for the bending moment  $M_U$  is then calculated from basic geometry as shown in Eq. (2.21). Subscript  $U$  is intentionally omitted in the components of the equation.

$$R_U = \frac{ABC}{4S} \quad (2.21)$$

$$S = \sqrt{p(p-A)(p-B)(p-C)}$$

$$p = \frac{A+B+C}{2}$$

The transformation from resisting bending moment inside the element into nodal forces is established through the Principle of Virtual Work. As exemplified in Figure 2.5, p. 13 **b**, the external work is produced as the force  $P$  initiates the displacement of node 1 by the amount  $\delta y$  along the negative  $y$ -direction. Consequently the angle between  $U$ -twines increases by  $\delta\alpha$ , producing the force couple  $M_U$  along the common edge and thus the external work. The counterclockwise rotation is positive in the sign convention for the right-hand side Cartesian coordinates. There are  $|V_3 - V_4|$  rows of  $U$  twines at the edge that contribute to the internal work. Equilibrium demands that the total work done is zero as shown in Eq. (2.22).

$$\delta W = \delta W_{int} + \delta W_{ext} = P\delta y + M_U(-\delta\alpha)V_{edge} \quad (2.22)$$

Diving both sides of the equation by infinitesimal displacement  $\delta y_1$  and letting it approach zero the expression for the external force  $P$  is obtained. In order to express the resisting force  $F_{y1}$  in Eq. (2.23) minus sign appears due to the equilibrium condition on the node  $P + F_{y1} = 0$ . Additionally, the zero contribution of  $V$ -twines is added to express the general formula.

$$F_{y1} = - \left( |V_3 - V_4| \frac{EI}{R_U} \frac{\partial \alpha}{\partial y} + |U_3 - U_4| \frac{EI}{R_V} \frac{\partial \beta}{\partial y} \right) \quad (2.23)$$

Applying the Eq. (2.23) to all nodes in all three directions yields the complete vector of 12 nodal forces due to the bending resistance Eq. (2.24).

$$\mathbf{F}^{bend} = - |V_3 - V_4| \frac{EI}{R_U} \frac{\partial \alpha}{\partial \mathbf{k}} - |U_3 - U_4| \frac{EI}{R_V} \frac{\partial \beta}{\partial \mathbf{k}} \quad (2.24)$$

where:

$$\mathbf{k} = (x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3, x_4, y_4, z_4)$$

The expressions for the gradient of the bending angle and the Jacobian of the force vector used for the Newton-Raphson method is found in Appendix A.

### Immersed weight

The weight of the netting immersed in water is taken into account in a straightforward manner as shown in Eq. (2.25). Here the volume of the netting is represented through the twine diameter  $D$ , twine length  $m_0$  and the amount of twines  $d$ . To avoid the choice between  $|\mathbf{U}|$  and  $|\mathbf{V}|$  the twine length is reasonably assumed to be  $m_0$ , because the strains in the twine are usually small. Therefore, all the Jacobian matrix entries of the weight force are zero, since nothing depends on the nodal positions of the triangle element. The direction of the force is negative if the netting is denser than the water ( $\rho_{net} > \rho_{water}$ ), otherwise it is positive.

$$\mathbf{F}_1^{weight} = \mathbf{F}_2^{weight} = \mathbf{F}_3^{weight} = \begin{pmatrix} 0 \\ 0 \\ \frac{1}{3}\pi\frac{D^2}{4}m_0d(\rho_{water} - \rho_{net})g \end{pmatrix} \quad (2.25)$$

### Drag on the netting

Unlike the axis-symmetric model, the 3D FEM model is able to take the drag forces on the netting into account. Those effects are simplified to a high extent, because the fully coupled fluid-structure interaction is still an open question. The fluid flow around the twines due to towing is characterized by the vector  $\mathbf{c} = (c_x, c_y, c_z)$  and assumed to be uniform and steady. The flow approaches the  $\mathbf{U}$  and  $\mathbf{V}$  twines with the angles of attack  $\alpha$  and  $\beta$  Eq. (2.26) as shown in Figure 2.6, p. 14.

$$\alpha = \arccos\left(\frac{\mathbf{c} \cdot \mathbf{U}}{|\mathbf{c}||\mathbf{U}|}\right) \quad \beta = \arccos\left(\frac{\mathbf{c} \cdot \mathbf{V}}{|\mathbf{c}||\mathbf{V}|}\right) \quad (2.26)$$

Hydrodynamic forces on the twines are calculated using the Morison equation assuming the twines have cylindrical shape. Due to the steady state assumption the inertia forces are zero, and the hydrodynamic forces are fully represented by the drag. The drag forces on the netting of the triangular element are represented by the pressure drag and the friction drag on  $U$  and  $V$  twines as shown in Figure 2.6, p. 14. Their magnitudes are presented in Eq. (2.27) and Eq. (2.28).

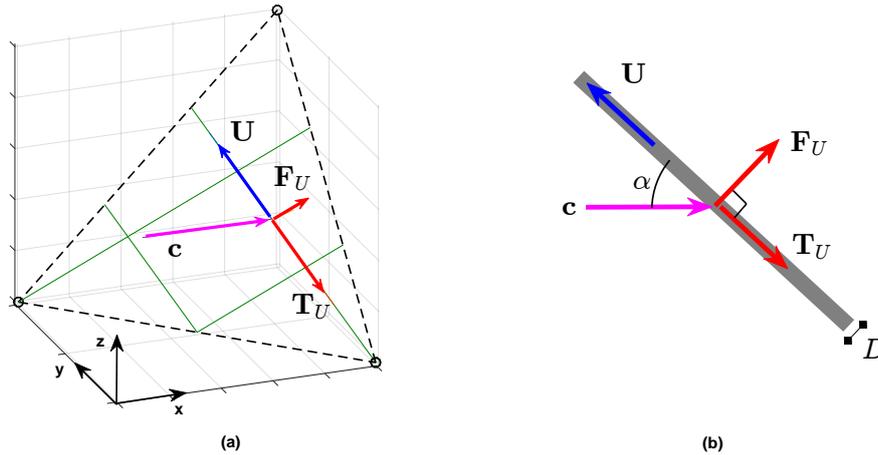


Figure 2.6: Drag force components on a  $U$  twine.

$$|\mathbf{F}_U| = \frac{1}{2}\rho_{water}C_dDm_0(|\mathbf{c}|\sin(\alpha))^2\frac{d}{2} \quad (2.27)$$

$$|\mathbf{T}_U| = C_f\frac{1}{2}\rho_{water}C_dDm_0(|\mathbf{c}|\cos(\alpha))^2\frac{d}{2} \quad (2.28)$$

The pressure drag and friction drag coefficients  $C_d$  and  $C_f$  for the twine is calculated based on its thickness  $D$  and the normal component of the current velocity  $c_n$  using Eq. (2.29) and Eq. (2.30) according to [21].

$$C_d = \begin{cases} \frac{8\pi (1 - 0.87s^{-2})}{Re_n s} & \text{if } 0 \leq Re_n \leq 1 \\ 1.45 + 8.55 Re_n^{-0.9} & \text{if } 1 < Re_n \leq 30 \\ 1.1 + 4 Re_n^{-0.5} & \text{otherwise} \end{cases} \quad (2.29)$$

$$C_f = \pi\mu \left( 0.55 Re_n^{0.5} + 0.084 Re_n^{\frac{2}{3}} \right) \quad (2.30)$$

where:

$$s = -0.077215665 + \log \frac{8}{Re_n} \quad (2.31)$$

$$Re_n = \frac{D \rho_{water} c_n}{\mu} \quad (2.32)$$

The pressure drag force is calculated as the pressure drag magnitude from Eq. (2.27) applied in the direction that is simultaneously perpendicular to the twine and to the plane of the triangular element. The perpendicular direction vector  $E$  is calculated as shown in Eq. (2.33).

$$\mathbf{E}_U = \mathbf{U} \times (\mathbf{c} \times \mathbf{U}) \quad \mathbf{E}_V = \mathbf{V} \times (\mathbf{c} \times \mathbf{V}) \quad (2.33)$$

$$\mathbf{F}_U = |\mathbf{F}_U| \frac{\mathbf{E}_U}{|\mathbf{E}_U|} \quad \mathbf{F}_V = |\mathbf{F}_V| \frac{\mathbf{E}_V}{|\mathbf{E}_V|} \quad (2.34)$$

The friction drag is calculated as the friction drag magnitude from Eq. (2.28) applied in the direction of the corresponding twine vector. Normalized cosines of  $\alpha$  and  $\beta$  account for the correct direction of the friction force.

$$\mathbf{T}_U = |\mathbf{T}_U| \frac{\cos(\alpha)}{|\cos(\alpha)|} \frac{\mathbf{U}}{|\mathbf{U}|} \quad \mathbf{T}_V = |\mathbf{T}_V| \frac{\cos(\beta)}{|\cos(\beta)|} \frac{\mathbf{V}}{|\mathbf{V}|} \quad (2.35)$$

Both forces are distributed uniformly over the element, therefore each node is affected by one third of the total force.

$$\mathbf{F}_1^{drag} = \mathbf{F}_2^{drag} = \mathbf{F}_3^{drag} = \frac{1}{3} (\mathbf{F}_U + \mathbf{F}_V + \mathbf{T}_U + \mathbf{T}_V) \quad (2.36)$$

The Jacobian of the force vector used for the Newton-Raphson method is found in Appendix A.

### Catch pressure

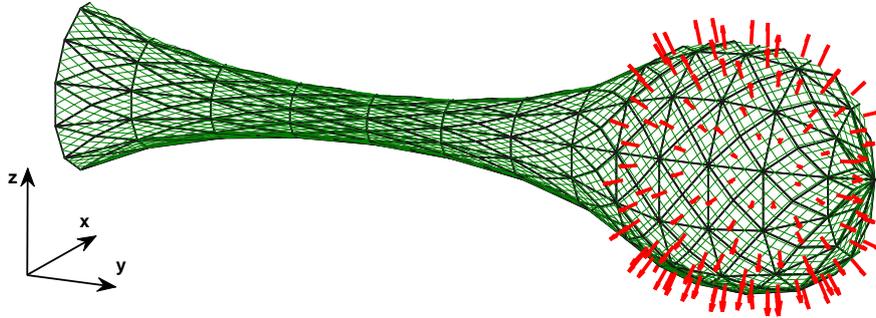
The catch is taken into account by subjecting triangular elements to the pressure Eq. (2.37) in the part where the netting is blocked by the catch. The  $C_d$  coefficient for the catch will vary along the cod-end, however, it was proved that it will reach its maximum over the rear of the cod-end and start to decrease over the region where the downstream component of the flow is small [22]. Therefore, a constant value between 1.2 and 1.4 can be assumed [11].

$$p = \frac{1}{2} \rho_{water} C_d |\mathbf{c}^2| \quad (2.37)$$

The pressure is assumed to be distributed uniformly over the triangular area, therefore each node is affected by one third of the resultant force as shown in Eq. (2.38).

$$\mathbf{F}_1^{catch} = \mathbf{F}_2^{catch} = \mathbf{F}_3^{catch} = \frac{\mathbf{S}_{12} \times \mathbf{S}_{13}}{2} \frac{p}{3} \quad (2.38)$$

Since the pressure is always pointing outside the cod-end it is important that the triangles have correct orientation. Node numbering must be counterclockwise when observing the elements from the outside of the cod-end to produce an outward normal as shown in Figure 2.7, p. 16. The Jacobian of the force vector used for the Newton-Raphson method is found in Appendix A.

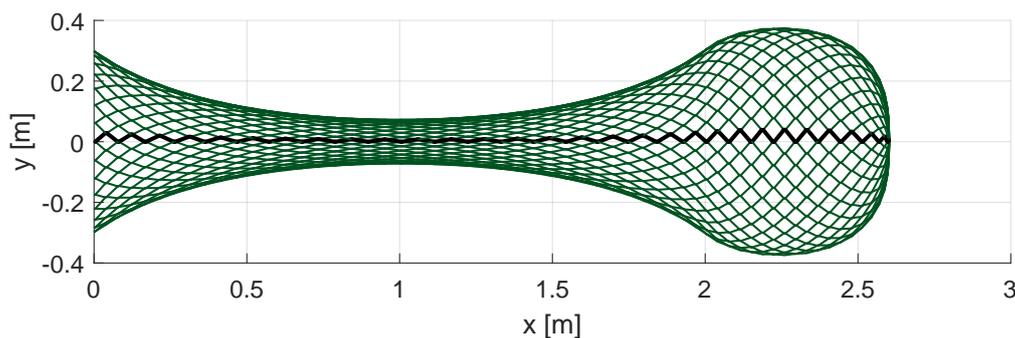


**Figure 2.7:** A cod-end partially blocked by catch. The normals show the direction of the catch pressure.

## 2.2 Axis-symmetric cod-end models

Numerical models assuming axis-symmetry were developed much later than the 3D FEM model. The models have been created for cod-ends with diamond, square, kite and hexagonal meshes [7] and [8]. This report considers only the model with the hexagonal meshes, since it can also simulate the diamond mesh cod-ends if the knot size is set to be very small. Being a reasonable assumption, the axis-symmetry limits the modelling freedom both in terms of geometry and forces in contrast to the previously described 3D FEM model. However, for some type of calculations, a more "lightweight" axis-symmetric model can achieve the same results, and thus will be more preferable than the 3D FEM model.

In the axis-symmetric models, the equilibrium shape of the cod-end can be obtained by considering the equilibrium of the nodes belonging to one row of twines, further called meridian (See Figure 2.8, p. 16). All other rows of twines in the cod-end circumference will have the same shape as the meridian due to the axis-symmetry assumption.



**Figure 2.8:** XY projection of the axis-symmetric T0 cod-end model with 30 meshes along, 30 meshes in circumference and 10 meshes blocked by catch. The black line represents the meridian row of twines.

In this project T0 and T90 models based on [7] are described. The meridian for two types of model is represented differently, therefore requires independent description. Both models rely on the following assumptions:

- Twines and knots are modelled as a 3D bar element, with only translation d.o.f. on both ends. The stiffness in bending and compression for the twines and knots is negligible. The material behaviour in tension is linear elastic.

- The only internal force involved in this model is the twine tension resistance. The only external force is the catch pressure. The drag force on the netting is considered negligible compared to the drag force on the catch.

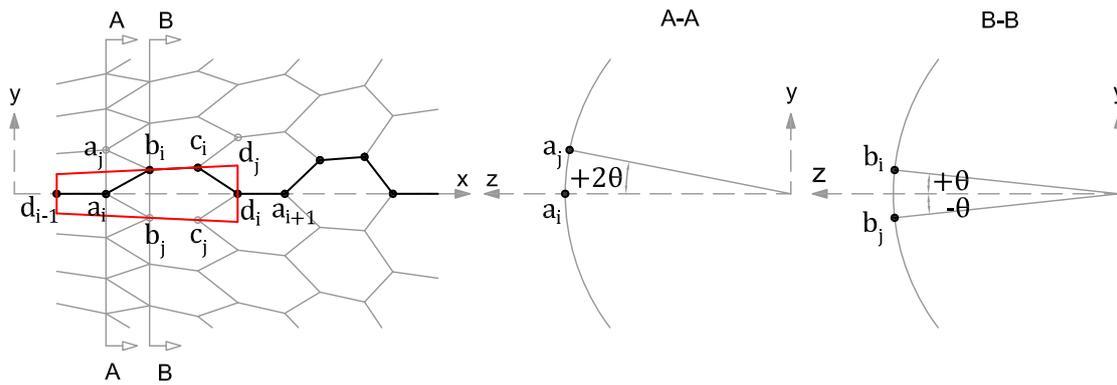
### 2.2.1 T0 model

#### Meridian position vector

The meridian for the T0 model is shown in Figure 2.9, p. 17. A part of meridian covering a mesh number  $i$  consists of 4 nodes:  $a_i, b_i, c_i, d_i$ . The node  $a_i$  and  $d_i$  always have  $y = 0$ , therefore they lay in the  $XZ$  plane. The node  $b_i$  and  $c_i$  lay in a radial plane, which creates angle  $\theta$  with the  $XZ$  plane. If the variable  $n_r$  denotes amount of meshes in the cod-end circumference, then the angle between two radial planes is calculated with Eq. (2.39).

$$\theta = \frac{\pi}{n_r} \quad (2.39)$$

The nodes  $a_j, b_j, c_j, d_j$  belong to the row of twines below and above the meridian. It is important to include these nodes to model the interaction of the meridian with the neighbouring rows of twines. In Figure 2.9, p. 17 sections A-A and A-B show, that the nodes with index  $j$  lay in the radial planes that are  $2\theta$  away from the planes of the meridian nodes.



**Figure 2.9:** Meridian with corresponding node pattern that repeats for every mesh. Mesh  $i$  is shown with red outline.  $ZY$  sections A-A and B-B are made through the nodes  $a_i$  and  $b_i$ .

The position of the aforementioned nodes can be represented in the cylindrical coordinates with  $x$  axial and  $r$  radial components. If the amount of meshes along is  $n_x$ , then the total amount degrees of freedom in the model  $n$  is  $2(1 + 4n_x)$ . The meridian position vector  $\mathbf{X}$  is defined as a collection of all the meridian nodes as shown in Eq. (2.40). There are 3 restrained degrees of freedom in the system. Known position and radius at the cod-end entrance give  $d_{0x} = 0$  and  $d_{0r} = r_0$ . Zero radius at the very end of the cod-end gives  $d_{nr} = 0$ .

$$\mathbf{X} = (d_{0x}, d_{0r}, a_{0x}, a_{0r}, \dots, a_{ix}, a_{ir}, b_{ix}, b_{ir}, c_{ix}, c_{ir}, d_{ix}, d_{ir}, \dots, d_{nx}, d_{nr})^T \quad (2.40)$$

Similarly, the total force vector is represented in the cylindrical coordinates as Eq. (2.41).

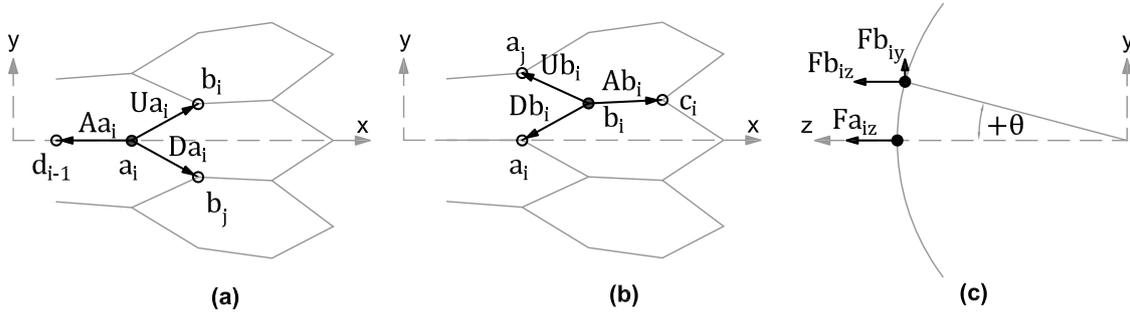
$$\mathbf{F} = (Fd_{0x}, Fd_{0r}, Fa_{0x}, Fa_{0r}, \dots, Fa_{ix}, Fa_{ir}, Fb_{ix}, Fb_{ir}, Fc_{ix}, Fc_{ir}, Fd_{ix}, Fd_{ir}, \dots, Fd_{nx}, Fd_{nr})^T \quad (2.41)$$

#### Twine tension

In order to create the vectors for twines and knots, each node is represented in the Cartesian coordinates as shown in Eq. (2.42). Notably, all  $i$  and  $j$  nodes can be represented using only the entries of the meridian position vector  $\mathbf{X}$ .

$$\begin{aligned}
a_i &= \begin{pmatrix} a_{ix} \\ 0 \\ a_{ir} \end{pmatrix} & a_j &= \begin{pmatrix} a_{ix} \\ a_{ir} \sin 2\theta \\ a_{ir} \cos 2\theta \end{pmatrix} & b_i &= \begin{pmatrix} b_{ix} \\ b_{ir} \sin \theta \\ b_{ir} \cos \theta \end{pmatrix} & b_j &= \begin{pmatrix} b_{ix} \\ -b_{ir} \sin \theta \\ b_{ir} \cos \theta \end{pmatrix} & (2.42) \\
c_i &= \begin{pmatrix} c_{ix} \\ c_{ir} \sin \theta \\ c_{ir} \cos \theta \end{pmatrix} & c_j &= \begin{pmatrix} c_{ix} \\ -c_{ir} \sin \theta \\ c_{ir} \cos \theta \end{pmatrix} & d_i &= \begin{pmatrix} d_{ix} \\ 0 \\ d_{ir} \end{pmatrix} & d_j &= \begin{pmatrix} d_{ix} \\ d_{ir} \sin 2\theta \\ d_{ir} \cos 2\theta \end{pmatrix}
\end{aligned}$$

There are 2 twine vectors and 1 knot vector emerging from each node of the meridian. Knot vectors are denoted  $\mathbf{A}$  for *axial* or *along*. Twine vectors are denoted  $\mathbf{U}$  for *upward* and  $\mathbf{D}$  for *downward*. The same notation remains true for rows of twines neighbouring to the meridian. The vectors are shown in Figure 2.10, p. 18 and defined in Eq. (2.43).



**Figure 2.10:** Twine and knot vectors for nodes  $a_i$  and  $b_i$ . The radial force for node  $b_i$  is a composition of  $y$  and  $z$  force components due to its position.

$$\begin{aligned}
\mathbf{Aa}_i &= \begin{pmatrix} d_{(i-1)x} - a_{ix} \\ 0 \\ d_{(i-1)r} - a_{ir} \end{pmatrix} & \mathbf{Ua}_i &= \begin{pmatrix} b_{ix} - a_{ix} \\ b_{ir} \sin \theta \\ b_{ir} \cos \theta - a_{ir} \end{pmatrix} & \mathbf{Da}_i &= \begin{pmatrix} b_{ix} - a_{ix} \\ -b_{ir} \sin \theta \\ b_{ir} \cos \theta - a_{ir} \end{pmatrix} \\
\mathbf{Ab}_i &= \begin{pmatrix} c_{ix} - b_{ix} \\ c_{ir} \sin \theta - b_{ir} \sin \theta \\ c_{ir} \cos \theta - b_{ir} \cos \theta \end{pmatrix} & \mathbf{Ub}_i &= \begin{pmatrix} a_{ix} - b_{ix} \\ a_{ir} \sin 2\theta - b_{ir} \sin \theta \\ a_{ir} \cos 2\theta - b_{ir} \cos \theta \end{pmatrix} & \mathbf{Db}_i &= \begin{pmatrix} a_{ix} - b_{ix} \\ -b_{ir} \sin \theta \\ a_{ir} - b_{ir} \cos \theta \end{pmatrix} \\
\mathbf{Ac}_i &= \begin{pmatrix} b_{ix} - c_{ix} \\ b_{ir} \sin \theta - c_{ir} \sin \theta \\ b_{ir} \cos \theta - c_{ir} \cos \theta \end{pmatrix} & \mathbf{Uc}_i &= \begin{pmatrix} d_{ix} - c_{ix} \\ d_{ir} \sin 2\theta - c_{ir} \sin \theta \\ d_{ir} \cos 2\theta - c_{ir} \cos \theta \end{pmatrix} & \mathbf{Dc}_i &= \begin{pmatrix} d_{ix} - c_{ix} \\ -c_{ir} \sin \theta \\ d_{ir} - c_{ir} \cos \theta \end{pmatrix} \\
\mathbf{Ad}_i &= \begin{pmatrix} a_{(i+1)x} - d_{ix} \\ 0 \\ a_{(i+1)r} - d_{ir} \end{pmatrix} & \mathbf{Ud}_i &= \begin{pmatrix} c_{ix} - d_{ix} \\ c_{ir} \sin \theta \\ c_{ir} \cos \theta - d_{ir} \end{pmatrix} & \mathbf{Dd}_i &= \begin{pmatrix} c_{ix} - d_{ix} \\ -c_{ir} \sin \theta \\ c_{ir} \cos \theta - d_{ir} \end{pmatrix} & (2.43)
\end{aligned}$$

When the twine and knot vectors are obtained it is possible to represent both direction and the magnitude of the nodal forces. This is exemplified in Eq. (2.44) for node  $a$ . Here  $m_0$  and  $l_0$  are the unstretched twine and knot lengths respectively.  $EA_l$  and  $EA_m$  represent twine and knot tensile stiffnesses. Similarly to Eq. (2.8), if the twine or the knot experiences compression, a significantly reduced stiffness value should be used.

$$\mathbf{Fa}_i = \frac{EA_l}{l_0} (|\mathbf{Aa}_i| - l_0) \frac{\mathbf{Aa}_i}{|\mathbf{Aa}_i|} + \frac{EA_m}{m_0} (|\mathbf{Ua}_i| - m_0) \frac{\mathbf{Ua}_i}{|\mathbf{Ua}_i|} + \frac{EA_m}{m_0} (|\mathbf{Da}_i| - m_0) \frac{\mathbf{Da}_i}{|\mathbf{Da}_i|} \quad (2.44)$$

The computation of the equilibrium position of the cod-end is done in the cylindrical coordinates. Nodal forces are transformed from the Cartesian coordinates, depending on whether a node is laying in the  $XZ$  plane or in the radial plane. It is seen from Figure 2.10, p. 18 c that for nodes  $a_i$  and  $b_i$  the radial component

of the force should be derived differently because of the inclination of the radial plane in the latter case. Generally, this transformation can be represented through the matrix  $\mathbf{T}(\theta)$  as shown in Eq. (2.45).

$$\begin{pmatrix} F_x \\ F_r \end{pmatrix} = \mathbf{T}(\theta) \begin{pmatrix} F_x \\ F_y \\ F_z \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \sin \theta & \cos \theta \end{bmatrix} \begin{pmatrix} F_x \\ F_y \\ F_z \end{pmatrix} \quad (2.45)$$

$$\theta = \begin{cases} 0 & \text{for nodes } a_i \text{ and } d_i \\ \frac{\pi}{n_r} & \text{for nodes } b_i \text{ and } c_i \end{cases}$$

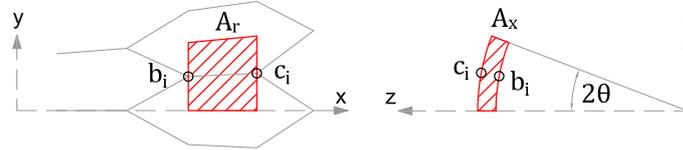
All nodal forces due to twine tension are added to the total nodal force vector Eq. (2.41). The elements of the Jacobian matrix are obtained by finding partial derivatives of the tensile forces with respect to the d.o.f. of the meridian vector  $\mathbf{X}$ . The Jacobian is found in Appendix B.

### Catch pressure

The pressure due to the hydrodynamic drag on the part of the netting blocked by catch is calculated in the same way as for the 3D model.

$$p = \frac{1}{2} C_d \rho_{water} |\mathbf{c}|^2$$

The axial and radial components of the hydrodynamic force acting on the meridian nodes can be computed by applying the pressure  $p$  to the relevant project areas. As shown in Figure 2.11, p. 19 a part of the cod-end between points  $b_i$  and  $c_i$  can be considered as a truncated cone with base radii  $b_{ir}$  and  $c_{ir}$ . The surface affecting the meridian is limited by the angle  $2\theta$  in the cod-end circumference. The projected area for the radial force is the whole surface, and the projected area for the axial force is a ring sector in the  $ZY$  plane. These areas are calculated in Eq. (2.47). The same procedure is applied for all consecutive pairs of the meridian nodes.



**Figure 2.11:** Projected surface areas  $A_r$  and  $A_x$  between nodes  $b$  and  $c$  for catch pressure distribution.

$$A_{b-c,x} = \frac{2\theta}{2\pi} \pi |b_{ir}^2 - c_{ir}^2| = \theta |b_{ir}^2 - c_{ir}^2| \quad (2.46)$$

$$A_{b-c,r} = (c_{ix} - b_{ix}) \frac{(2\theta b_{ir} + 2\theta c_{ir})}{2} = \theta (c_{ix} - b_{ix}) (b_{ir} + c_{ir}) \quad (2.47)$$

The pressure  $p$  acting on the surface between nodes  $b_i$  and  $c_i$  is distributed evenly, so that each node is affected by the half of the resultant force on the surface. Node  $b_i$  also creates surface with the node  $a_i$ . Therefore, the force calculation for the node  $b_i$  includes contributions from both preceding and successive surfaces as shown in Eq. (2.48) and Eq. (2.49). Exceptions are the very first and very last nodes affected by catch pressure, since they only have 1 neighbouring node affected by the catch pressure.

$$F b_x = \theta |a_{ir}^2 - b_{ir}^2| \frac{P}{2} + \theta |b_{ir}^2 - c_{ir}^2| \frac{P}{2} \quad (2.48)$$

$$F b_r = \theta (b_{ix} - a_{ix}) (a_{ir} + b_{ir}) \frac{P}{2} + \theta (c_{ix} - b_{ix}) (b_{ir} + c_{ir}) \frac{P}{2} \quad (2.49)$$

All nodal forces due to catch pressure are added to the total nodal force vector Eq. (2.41). The elements of the Jacobian matrix are obtained by finding partial derivatives of the catch pressure forces with respect to the d.o.f. of the meridian vector  $\mathbf{X}$ . The Jacobian is found in Appendix B.

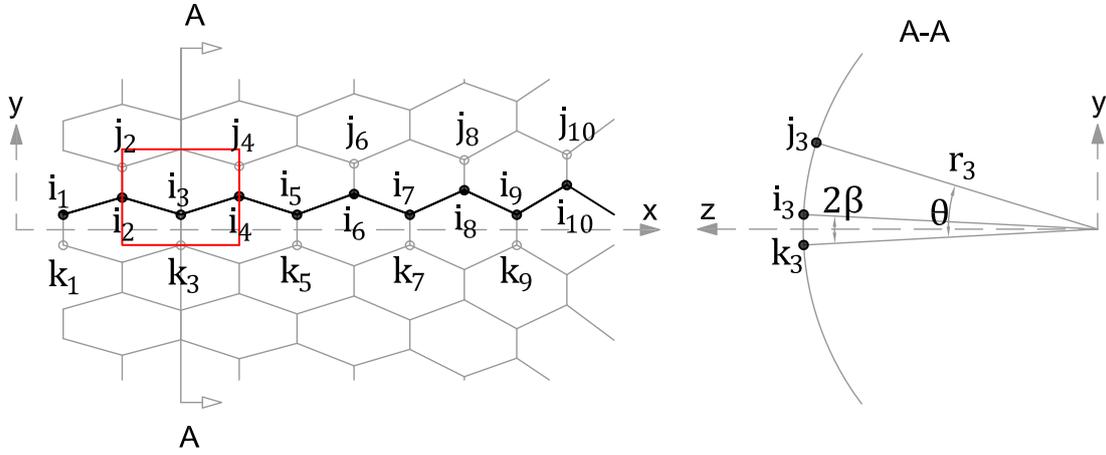
## 2.2.2 T90 model

### Meridian position vector

The meridian for the T90 model is shown in Figure 2.12, p. 20. The nodes of the meridian are denoted  $i$ . The meridian will interact with the neighbouring rows of twines. The nodes of the twine row above the meridian are denoted  $j$  and row below  $k$ . The meridian is linked to the row  $j$  through the nodes with even indexes and to the row  $k$  through the nodes with odd indexes. In the T90 model the angle  $\theta$  is redefined, and now represents an angle that one full mesh covers in the cod-end circumference as shown in Eq. (2.50).

$$\theta = \frac{2\pi}{n_r} \quad (2.50)$$

The plane  $XZ$  with  $y = 0$  is chosen to be between the meridian  $i$  and row  $k$ . In this manner the symmetry is abused and the coordinates of  $k$  nodes are the coordinates of  $i$  nodes with the negative  $y$  component. The coordinates of  $j$  nodes are defined as clockwise rotation of  $k$  nodes by the mesh angle  $\theta$ . This applies for nodes with both even and odd indexes.



**Figure 2.12:** Meridian with corresponding node pattern that repeats at every mesh. A mesh is shown with red outline.  $ZY$  section passes through nodes  $i_3$ ,  $j_3$  and  $k_3$ . For rows  $j$  and  $k$ , only nodes linked to the meridian are labeled.

The position of the aforementioned nodes is represented in the Cartesian coordinates. If the amount of meshes along is  $n_x$ , then the total amount d.o.f. in the model  $n$  is  $3(1 + 2n_x)$ . The meridian position vector  $\mathbf{X}$  is defined as a collection of all the meridian nodes as shown in Eq. (2.51). There are 5 restrained degrees of freedom in the system. Known position and radius at the cod-end entrance give  $i_{1x} = 0$ ,  $i_{1y} = \frac{l_0}{2}$  and  $i_{1z} = \sqrt{r_0^2 - i_{1y}^2}$ . Zero radius at the very end of the cod-end gives  $i_{ny} = i_{nz} = 0$ .

$$\mathbf{X} = (i_{1x}, i_{1y}, i_{1z}, i_{2x}, i_{2y}, i_{2z}, \dots, i_{mx}, i_{my}, i_{mz})^T \quad (2.51)$$

Similarly, the total force vector is represented in the Cartesian coordinates as shown in Eq. (2.52).

$$\mathbf{F} = (Fi_{1x}, Fi_{1y}, Fi_{1z}, Fi_{2x}, Fi_{2y}, Fi_{2z}, \dots, Fi_{mx}, Fi_{my}, Fi_{mz})^T \quad (2.52)$$

### Twine tension

The definition of the node positions is given in the Cartesian coordinates in terms of the meridian position components  $\mathbf{X}$  in Eq. (2.53).

$$i_4 = \begin{pmatrix} i_{4x} \\ i_{4y} \\ i_{4z} \end{pmatrix} \quad k_4 = \begin{pmatrix} i_{4x} \\ -i_{4y} \\ i_{4z} \end{pmatrix} \quad j_4 = \begin{pmatrix} i_{4x} \\ i_{4z} \cos \theta + i_{4y} \sin \theta \\ i_{4z} \sin \theta - i_{4y} \cos \theta \end{pmatrix} \quad (2.53)$$

Similarly to the T0 model, there are 2 twine vectors and 1 knot vector emerging from each node of the meridian. Knot vectors are denoted  $\mathbf{A}$  for *around*. Twine vectors are denoted  $\mathbf{B}$  for *backward* and  $\mathbf{F}$  for *forward*. The same notation remains true for rows of twines neighbouring to the meridian. The definition of vectors emerging from the odd and even nodes of the meridian is different because the meridian nodes are linked to either  $k$  or  $j$  as shown in Eq. (2.13). As an example, the vectors emerging from a node with even index  $i_4$  and a node with odd index  $i_5$  are presented in Eq. (2.54) and Eq. (2.55) respectively.

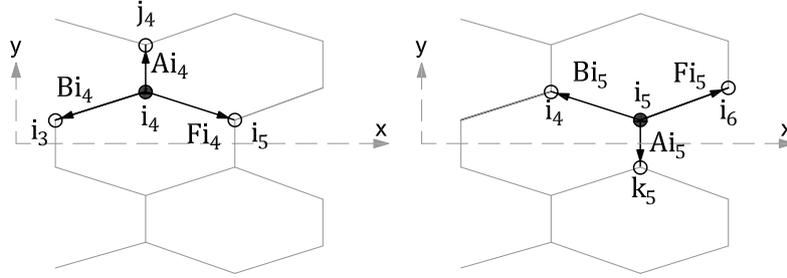


Figure 2.13: F

$$\mathbf{A}i_4 = \begin{pmatrix} 0 \\ i_{4z} \cos \theta + i_{4y} \sin \theta - i_{4y} \\ i_{4z} \sin \theta - i_{4y} \cos \theta - i_{4z} \end{pmatrix} \quad \mathbf{B}i_4 = \begin{pmatrix} i_{3x} - i_{4x} \\ i_{3y} - i_{4y} \\ i_{3z} - i_{4z} \end{pmatrix} \quad \mathbf{F}i_4 = \begin{pmatrix} i_{5x} - i_{4x} \\ i_{5y} - i_{4y} \\ i_{5z} - i_{4z} \end{pmatrix} \quad (2.54)$$

$$\mathbf{A}i_5 = \begin{pmatrix} 0 \\ 2i_{4y} \\ 0 \end{pmatrix} \quad \mathbf{B}i_5 = \begin{pmatrix} i_{4x} - i_{5x} \\ i_{4y} - i_{5y} \\ i_{4z} - i_{5z} \end{pmatrix} \quad \mathbf{F}i_5 = \begin{pmatrix} i_{6x} - i_{5x} \\ i_{6y} - i_{5y} \\ i_{6z} - i_{5z} \end{pmatrix} \quad (2.55)$$

The tensile forces along those vectors are collected in the resulting nodal force as exemplified in Eq. (2.56) on the node  $i_4$ .

$$\mathbf{F}i_4 = \frac{EA_l}{l_0} (|\mathbf{A}i_4| - l_0) \frac{\mathbf{A}i_4}{|\mathbf{A}i_4|} + \frac{EA_m}{m_0} (|\mathbf{F}i_4| - m_0) \frac{\mathbf{F}i_4}{|\mathbf{F}i_4|} + \frac{EA_m}{m_0} (|\mathbf{B}i_4| - m_0) \frac{\mathbf{B}i_4}{|\mathbf{B}i_4|} \quad (2.56)$$

All nodal forces due to twine tension are added to the total nodal force vector Eq. (2.52). The elements of the Jacobian matrix are obtained by finding partial derivatives of the tensile forces with respect to the d.o.f. of the meridian vector  $\mathbf{X}$ . The Jacobian is found in Appendix B.

### Catch pressure

The forces on the netting blocked by the catch are calculated in the manner as for the T0 model. However, as shown in Eq. (2.50) for the T90 model the angle  $\theta$  was redefined, and now corresponds to the part of circumference covered by the entire mesh. Now, the surface of the truncated cone that affects a meridian node corresponds to  $\frac{1}{2}\theta$  instead of  $2\theta$  as shown in Figure 2.14, p. 21.

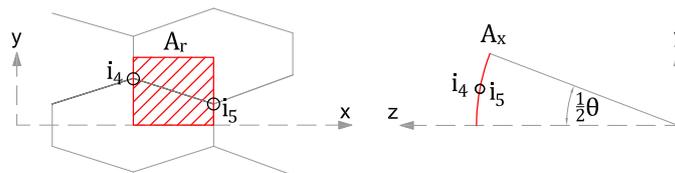


Figure 2.14: Projected surface areas  $A_r$  and  $A_x$  between nodes  $i_4$  and  $i_5$  for catch pressure distribution. Notably  $A_x = 0$  in this case, because there is no change in radius between the nodes.

The formulas for the projected area from Eq. (2.47) are updated as exemplified in Eq. (2.59).

$$A_{4-5,x} = \frac{\frac{1}{2}\theta}{2\pi} \pi |i_{5r}^2 - i_{4r}^2| = \frac{\theta}{4} |i_{5r}^2 - i_{4r}^2| \quad (2.57)$$

$$A_{4-5,r} = (i_{5x} - i_{4x}) \frac{(\frac{1}{2}\theta i_{4r} + \frac{1}{2}\theta i_{5r})}{2} = \frac{\theta}{4} (i_{5x} - i_{4x}) (i_{4r} + i_{5r}) \quad (2.58)$$

$$\text{where } i_r = \sqrt{i_y^2 + i_z^2} \quad (2.59)$$

The axial and radial components of the hydrodynamic force at node  $i_5$  are calculated accordingly, taking into account the contributions from both neighbouring reference areas as shown in Eq. (2.60) and Eq. (2.61).

$$Fi_{5x} = \theta |i_{4r}^2 - i_{5r}^2| \frac{P}{8} + \theta |i_{5r}^2 - i_{6r}^2| \frac{P}{8} \quad (2.60)$$

$$Fi_{5r} = \theta (i_{5x} - i_{4x}) (i_{4r} + i_{5r}) \frac{P}{8} + \theta (i_{6x} - i_{5x}) (i_{5r} + i_{6r}) \frac{P}{8} \quad (2.61)$$

The force vector is required to be in the Cartesian coordinates, therefore the forces have to be transformed. The angle  $\beta$  is shown in Figure 2.12, p. 20.

$$Fi_{5y} = Fi_{5r} \sin \beta = Fi_{5r} \frac{i_{5y}}{i_{5r}} \quad (2.62)$$

$$Fi_{5z} = Fi_{5r} \cos \beta = Fi_{5r} \frac{i_{5z}}{i_{5r}} \quad (2.63)$$

All nodal forces due to the catch pressure are added to the total nodal force vector Eq. (2.52). The elements of the Jacobian matrix are obtained by finding partial derivatives of the catch pressure forces with respect to the d.o.f. of the meridian vector  $\mathbf{X}$ . The Jacobian is found in Appendix B.

# Computer implementation and solution algorithm

---

In this chapter, a description of the computer implementation of the two numerical models is given. Usually, books and research papers presenting numerical models focus mostly on the underlined assumptions and mathematical formulations, leaving the actual implementation to the reader. This approach is completely reasonable, however, it in some cases can slow down or prevent the reader from transferring his understanding of the model to actual results. An example for that could be the information on the appropriate initial position for the Newton-Raphson iterative method. An initial shape is a key element of the method, and while a good one can make the scheme converge in very few iterations, a bad one might not converge at all. Sharing an unsuccessful implementation experience is also useful, since the reader can either avoid it, or try to approach it from a different perspective. Therefore, in this chapter some of the most important implementation choices are supplied by the numerical tests and comparison with other possible solutions.

Two languages used in the current work are C# and MATLAB. An analyst or researcher naturally wants to work with code which is easily extensible towards future demands, easily maintainable, but still efficient and portable across many platforms, which well matches the key qualities of object-oriented programming (OOP) paradigm. C# is a modern general-purpose OOP language that was chosen as a main development language. Apart from all the advantages of .NET Framework for general software development, it lacks a standard linear algebra library that is of a great importance for numerical modelling. Therefore, a choice of the appropriate linear algebra library is discussed later in the chapter. As a secondary language MATLAB was chosen. MATLAB is multi-paradigm scientific computing environment widely used at AAU. It was chosen for fast prototyping and data visualization, because of its extensive numerical libraries and toolboxes.

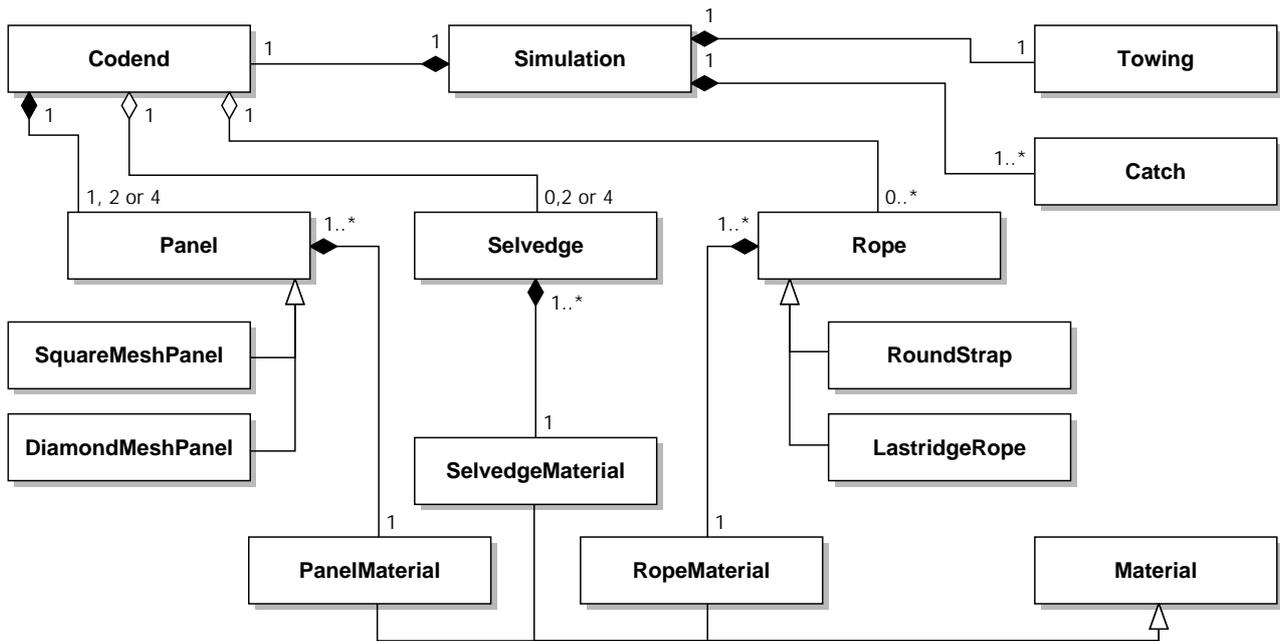
## 3.1 Program structure and classes' description

In OOP paradigm the key component is a class that models the state and behavior of a real-world object. Classic approach to modelling in OOP is by using Unified Modeling Language (UML) diagrams that represent the static view of an application. They describe what attributes, operations and relations to other classes must be present in the system being modeled [23]. The design of the classes and their relationships is not straightforward and there is no "best" solution. Current class relationships were established from the perspective of real life-objects like cod-end, panel, etc. Working with those classes is intuitive for the fishing industry. However, since it is a FEM modelling program, it could also have been approached from the perspective of the FEM such as domain, element, etc. This structure would favour the users with structural mechanics background.

### 3.1.1 3D model

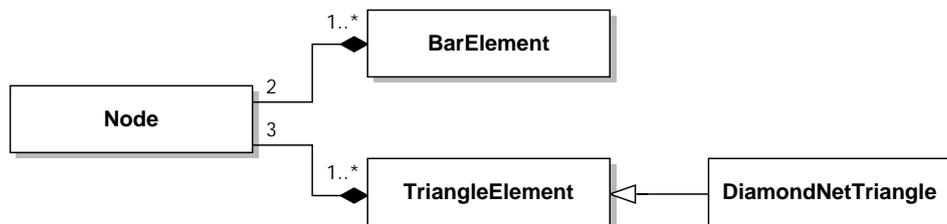
The class diagram of the 3D model is presented in Figure 3.1, p. 24. The central class of the program is **Codend** that contains a list of panels, and can also contain list of selvages and ropes. **Panel** class generalizes different types of panels such as a panel with square meshes, diamond meshes, and allows extension for hexagonal meshes in the future. **Rope** class is a generalization of classes with rope/cable like behaviour, that are modelled in the program as a group of simple bar elements. From the modelling perspective **Selvedge** class should be a special case of **Rope** since it is also modelled with the bar elements. However, since it is not an actual rope in reality, it was decided to treat it as a separate class. **Panel**, **Selvedge** and **Rope** classes contain methods that are used by **Codend** class to assemble and discretize the cod-end structure and create lists of nodes, bars, triangles and boundary conditions for the finite element calculations. The FEM

---



**Figure 3.1:** Class diagram for the 3D model. Attributes and methods are intentionally hidden and only class relations are shown.

classes are shown in Figure 3.2, p. 24. **Node** class contains a spatial coordinates of a finite element node. **BarElement** and **TriangleElement** contain the methods to obtain nodal forces and tangent stiffness matrix for the particular element as described in Section 2.1, p. 7. The classes also contain the information of the element d.o.f. for later global system assembly. **TriangleElement** holds references to its vertex node objects and also  $uv$  coordinates. Furthermore, **TriangleElement** is a generalization of an element based on diamond netting, and can be later extended to the hexagonal netting.



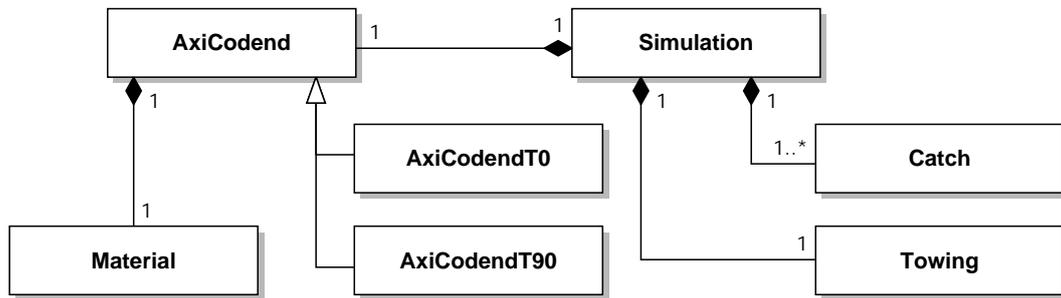
**Figure 3.2:** Sub-diagram for the finite elements of the 3D model. The finite elements are according to [11]. Attributes and methods are intentionally hidden and only class relations are shown.

When the panels are first discretized in 2D, the **Panel** class is capable of performing the geometric manipulations in order to shape the initial position for the calculation. Those manipulations include rotation, translation, scaling and mapping to a 3D body of revolution such as a cylinder or a shape obtained from the axis-symmetric model.

An essential input to **Panel**, **Selvedge** and **Rope** classes is the corresponding **Material** class, that contains the information that dictates the mechanical behaviour of a corresponding triangular or bar element. It is, for example, mesh opening angle at rest, twine/rope thickness, various stiffness parameters, etc.

**Simulation** class contains the functionality in order to simulate the towing of a structure represented by **Codend** with various amount of catch inside. Classes **Towing** and **Catch** keep the information required to calculate the twine drag and catch pressure forces on the model. The finite element lists passed by **Codend** class are used to assemble forces and stiffness for each step in the numerical calculation. The class implements Newton-Raphson iteration method, that can be manipulated with a helper class **SolverSettings** (not shown in the diagram).

### 3.1.2 Axis-symmetric model



**Figure 3.3:** Class diagram for the axis-symmetric model. Attributes and methods are intentionally hidden and only class relations are shown.

The axis-symmetric model is implemented separately from the 3D model. Due to limited modelling options the class diagram for the model is simpler as shown in Figure 3.3, p. 25. As previously, the central class of the program is **AxiCodend** which is a generalization of the T0 and the T90 cod-end models. It was decided not to rely on FEM classes as in 3D model. It is due to the fact that the elements represent real twines and knots, so the discretization is not required. There are only bar elements with 2 different properties that are assembled in a well known pattern to form the meridian. Therefore global forces and tangent stiffness are calculated directly as a state variables of the **AxiCodend** depending on the meridian position. **AxiCodend** also provides an initial shape that is discussed later in the chapter. **Simulation** class is similar to the one of the 3D model.

### 3.1.3 Input-output

Current version of the program does not have a user interface. Inputs for the model are kept in a text file, that are read by the program on execution. There are essential inputs and optional inputs. Essential are, for example, the number of meshes along a panel of the cod-end, panel material, towing speed, etc. Optional outputs are the advanced parameters of the solver such as convergence tolerance. Output cod-end shape and results for catch volume, towing resistance, etc. also come in text files and can be read and visualized in MATLAB .

## 3.2 Meshing and assembly of 3D cod-end model

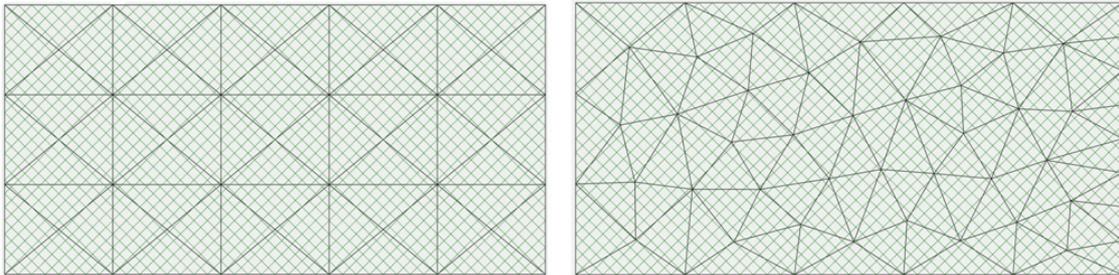
To prepare the model for the calculation, individual panels selvages and round straps have to be merged together into a structure and discretized in the consistent way.

As mentioned before **Panel** class contains methods for subdividing (meshing) it into triangular elements. The mesh can be categorized depending on its generation principle. In a structured mesh, a repeated pattern is followed where every node is uniquely defined according to a specific algorithm. In an unstructured mesh all nodes are typically positioned arbitrarily and then triangles are produced by triangulation refinement and smoothing. In this case no regular pattern is followed [24].

There are many advantages of structured mesh compared to the unstructured mesh. It provides full control over vertex location and number of triangles. Furthermore, there is better control over the vertex indices and thus, the ability to reduce the band width of the finite element stiffness matrix with less effort already on the stage of the discretization. In contrast, during the generation of an unstructured mesh the vertices are typically added and deleted randomly (unless constrained) in order to achieve the imposed requirements of minimum angle or maximum triangle area.

Structured mesh allows for easier neighbouring vertex or triangle search in an algorithmic way without need of a mesh data structure containing neighbour information. However, if the structured mesh should be refined or/and smoothed, the mesh data structure containing various triangle and edge connectivity information is required.

Structured mesh can be aligned with the forces or with the geometric features of the structure. For the cod-end it is beneficial that the triangles are aligned both along and across the cod-end panels as shown in



**Figure 3.4:** Structured (left) and unstructured (right) meshing of a netting panel. Unstructured mesh is created with constrained Delaunay triangulation using TRIANGLE.NET library for C# [25].

Figure 3.4, p. 26. This alignment gives the opportunity for the triangles to bend over the edges by the same amount providing the same curvature.

The biggest advantage of the unstructured meshes is the ability to mesh complex 3D free form geometries. In addition, the same algorithms that create the mesh can also be used to optimize it to improve the mesh quality. However, this advantage might not be critical for the cod-end modelling, since geometry of a cod-end is to some extent predictable. Moreover, it is not required to mesh the entire cod-end right away in 3D, because it consists of panels that each can be meshed in 2D and then mapped to a 3D shape. Therefore it was decided to implement only the structured mesh generation in the program.

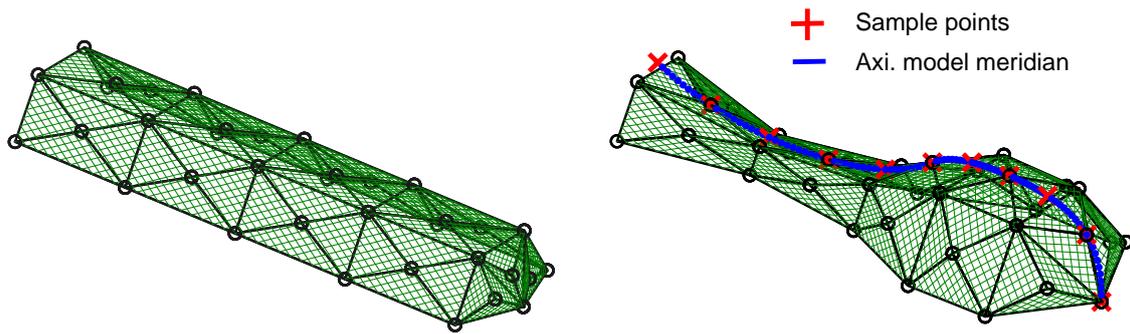
The discretization and assembly of the cod-end follows the following pattern:

1. Individual panels are discretized. Initially the panel lies in the  $XY$  plane and is 1 by 1 unit size. For each panel a structured grid of rectangles is created according to the mesh resolution information stored in **MeshSettings** class. Then, a central point is added to each rectangle to obtain 4 triangles. Afterwards, the unit panel is used to create the correct mesh coordinates and the Cartesian coordinates by scaling. Triangular elements are now created with the correct coordinates in the two systems. First and last row of the panel are labeled for later application of boundary conditions represented in **Restraint** class.
2. In the current version of the program selvedge and strap discretization follows the discretization of corresponding panels. Nodes on the longitudinal edges are labeled and assigned to selvedges. Nodes across the panel laying at the specific offset from the start are labeled and assigned as round strap nodes.
3. Initial shape is assembled through translation, rotation and bending (mapping to a body of revolution) of the flat panels that are implemented in **Panel** class. Current version of the program supports an initial shape based on a cylinder, or a pre-calculated solution of an axis-symmetric model as shown in Figure 3.5, p. 27. The effect of the initial shapes on the program performance is studied in more details later in the chapter.
4. When the cod-end shape is assembled it is crucial to merge individual meshes and avoid copies of the nodes that should represent the same d.o.f. along the edges and the end of the cod-end. Only unique nodes are left and the node references stored in triangles are updated. It is important to notice that while a copy of a node can be removed, the consistency in  $uv$  coordinates is preserved, since they are the attributes of the triangles.

After this process **Codend** class has created consistent and unique FEM lists **List<Node>**, **List<BarElement>**, **List<TriangleElement>** and **List<Restraint>** to be used in **Simulation** class for the equilibrium shape calculation.

### 3.3 Choice of mathematical library

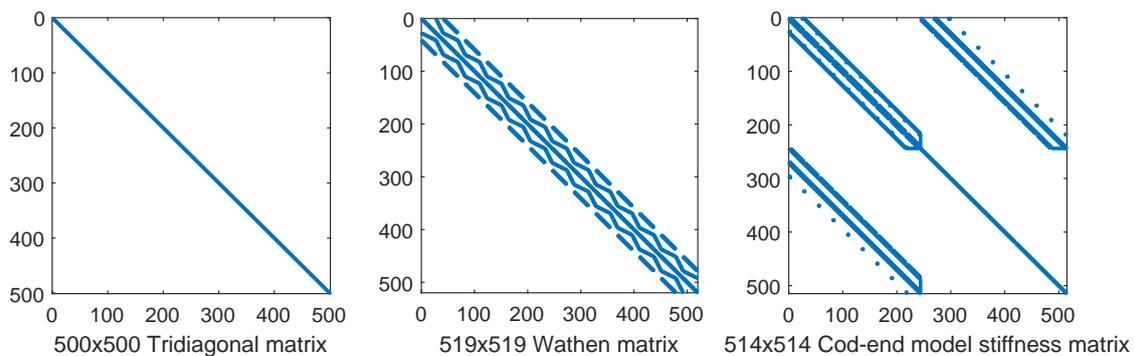
Typically, iteration schemes for non-linear systems of equations spend most of their time solving the linear system of equation, therefore, a good mathematical library is the biggest limiting factor that influences the program performance. Problem at hand is a finite element problem where the coefficient matrix is usually sparse (having significantly more zero entries than non-zero entries). Therefore, the priority of the search was given to libraries with direct sparse matrix solver or a fast implementation of a direct dense matrix solver.



**Figure 3.5:** A top panel with a coarse structured mesh from Figure 3.4, p. 26 can be mapped to a cylinder with entrance radius or to a shape pre-calculated with the axis-symmetric model. For the axis-symmetric model the sample points are obtained depending on the mesh discretization settings.

The goal of this chapter is to evaluate two candidates found among open source libraries. The first candidate is MATH.NET NUMERICS, is an extensive mathematics library, written in C# [26]. Although, the library is still under development and only has dense solvers, it is designed in such way, that performance-sensitive algorithms can be swapped with alternative implementations by the concept of providers. Providers become useful, when they can leverage a platform-native (C++ or FORTRAN) high performance library, like INTEL MKL instead of the default purely managed (C#) provider. The second candidate is CSPARSE.NET, a concise library for solving sparse linear systems with direct methods written in C# [27].

Two libraries are tested on 3 different types of  $N \times N$  matrices with 5 sizes  $N$  spanning from 300 to 8000 degrees of freedom. Matrix 1 is a tridiagonal matrix, that arises in many numerical solutions for example Laplace equation. Test matrix 2 is a Wathen matrix, which is a classic test case for finite element method. Test matrix 3 is a tangent stiffness matrix for the 3D cod-end model that arises after structured meshing discussed earlier. The sparsity pattern of the test matrices is presented in Figure 3.6, p. 27.



**Figure 3.6:** Sparsity pattern of the three test matrices with approximately 500 d.o.f.

It is seen that three test matrices differ in their bandwidth, and it is expected the sparse solver to be influenced by this fact. On the other hand, dense solver is expected to calculate solutions in the same time. The results of the performance test are presented in Figure 3.7, p. 28.

It is seen that the band width of the matrix is highly influential for the managed sparse solver. Nevertheless, in all cases the managed sparse solver outperforms the dense solver, even with a much more efficient native implementation. In some cases the difference is counted in orders of magnitude. The efficiency of CSPARSE.NET in 3D model can be further improved by optimizing the bandwidth of the stiffness matrix.

Especially big speed up can be achieved by using CSPARSE.NET for the tridiagonal matrix with a very narrow band. The structure of the Jacobian in axis-symmetric model is very similar and has a bandwidth of 6 or 9 depending on whether it is the T0 or the T90 model (Appendix B). For the brief comparison, the final axis-symmetric model code was able to perform 91 simulations of a 100 mesh long cod-end filled by catch consecutively blocking meshes 10 to 100 in only 5 seconds. In contrast, the same simulation with MATH.NET NUMERICS took approximately 129 seconds. As a remark, other attempted libraries with

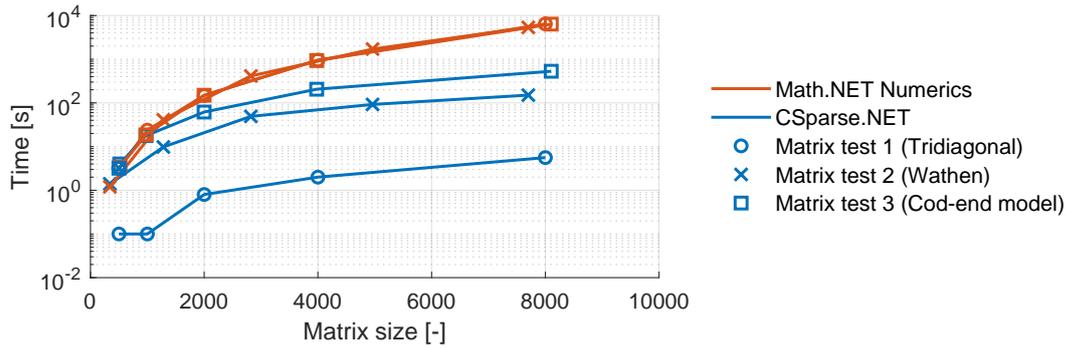


Figure 3.7: Performance comparison of the solution of 3 different linear systems of equation.

purely managed dense solvers such as ACCORD.NET[28] were not shown in this comparison, since their performance is orders of magnitude slower than those presented in the section.

### 3.4 Solution to non-linear system

The equilibrium shape of the cod-end structure is obtained by solving the system of non-linear equilibrium equations Eq. (3.1) by the methods implemented in **Simulation** class. Current position vector  $\mathbf{X}_i$  in both models is a collection of all d.o.f. of the system.

$$\mathbf{F}(\mathbf{X}_i) = \mathbf{0} \quad (3.1)$$

The system can be solved with Newton-Raphson (NR) iteration:

$$\mathbf{h}_i = -\mathbf{J}^{-1}(\mathbf{X}_i) \mathbf{F}(\mathbf{X}_i) \quad (3.2)$$

$$\mathbf{X}_{i+1} = \mathbf{X}_i + \lambda \mathbf{h}_i \quad (3.3)$$

Where  $\mathbf{J}$  is the Jacobian matrix of the force vector and also negative of the tangent stiffness matrix. Power  $-1$  indicates that the linear system of equations has to be solved for vector  $\mathbf{h}_i$  rather than inverting the  $\mathbf{J}$  which is never done in practice. One iteration of the solution algorithm consists of 3 main parts:

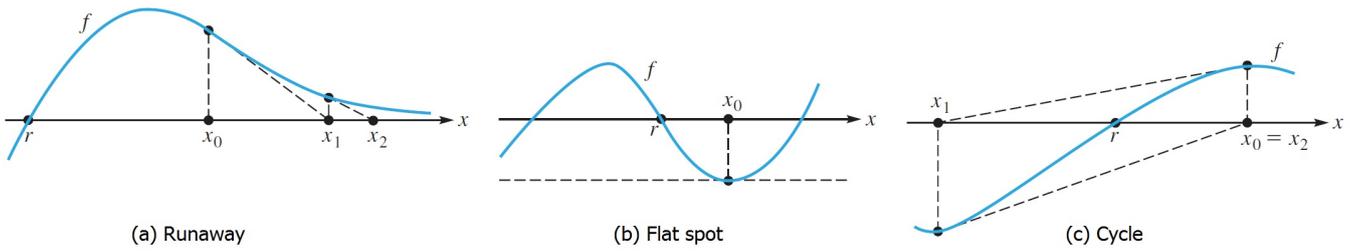
1. Assembly of the global force vector  $\mathbf{F}(\mathbf{X}_i)$  global and tangent stiffness matrix  $-\mathbf{J}(\mathbf{X}_i)$  corresponding to current deformed configuration  $\mathbf{X}_i$ .

For the 3D model, local force vectors and stiffness matrices for each element are obtained using methods implemented in **BarElement** and **TriangleElement** according to Section 2.1, p. 7 and assembled in parallel into their global systems. Parallel assembly improves the performance of the iteration scheme by taking advantage of the fact that final global system does not depend on assembly sequence. In the axis-symmetric model the global system skips the creation of the local force vectors and stiffness matrices and is filled with values right away due to simplicity of the structure. In both models, the result of the assembly is a sparse matrix. In both models the boundary conditions are taken into account during the assembly by assigning 1 to restrained d.o.f. and nullifying the rest of the entries in the row and column corresponding to this d.o.f.

2. Solution of the linear system of equations for the displacement vector  $\mathbf{h}_i$  is performed with a direct sparse solver presented earlier.
3. Update of the previous deformed configuration to obtain  $\mathbf{X}_{i+1}$ .

It is known that the classic NR method is not a globally convergent algorithm [29]. A globally convergent algorithm is understood as an algorithm with the property that for any initial iterate the iteration either converges to a solution or fails to do so in one of a small number of ways. The classic NR suffers from the problems like divergence, flat spot or cycling presented in Figure 3.8, p. 29.

Cases (a) and (b) are due to the poor choice of the initial position, that has to be *close enough* to the solution for algorithm to converge. In case (c) the full Newton step ( $\lambda = 1$ ) is attempted toward the root, but



**Figure 3.8:** Cases when classic NR iteration fails - an example with 1 non-linear equation. [20]

the length of step is too large, and overshooting happens. The algorithm is improved by introducing a line search algorithm along Newton direction for deciding on a NR step length  $\lambda$  that guarantees the reduction of the force residual by certain small value  $\beta$ . Chosen line search algorithms are 3-point safeguarded parabolic line search and Armijo Rule [18] are described in Appendix C.

In many cases  $\mathbf{J}$  is ill-conditioned and non-symmetric positive definite. The matrix becomes ill-conditioned when twine stiffness is reduced due to compressed/slack twines and there is orders of magnitude difference between the Jacobian entries. Another case is when the bending forces are included, that usually are orders of magnitude smaller than the tensile forces [18]. In addition, the parts of  $\mathbf{J}$  corresponding to non-conservative forces such as drag, are not symmetric, and therefore  $\mathbf{J}$  loses symmetric positive definiteness (SPD) property that can result in NR iteration calculating a direction that is not a descend direction [30].

A classic workarounds to improve these issues suggests to add an arbitrary value  $\alpha$  along the diagonal of the stiffness matrix that is between 0 and  $\|\mathbf{J}\|$  [31]. It is effective to decrease this added diagonal stiffness value along the NR iterations as better equilibrium position are found in order to utilize quadratic convergence rate of NR in the proximity of the solution. However, it is not trivial to choose an optimal added stiffness  $\alpha$  and a decrease factor for each problem, therefore those are available as advanced parameters in **SolverSettings** class. Sometimes the initially chosen  $\alpha$  might not provide the sufficient stability to the algorithm, even with line search techniques. In that case the iteration scheme either diverges or converges to an incorrect solution. Although a good initial shape can greatly minimize dependency on a good  $\alpha$  value, it is still non-negligible. To prevent that, NR method is restarted with higher  $\alpha$  as soon as the divergence or an incorrect (tangled) solution is detected. The pseudo-code summarizing all those modifications to classic NR is shown in Algorithm 1, p. 30. The algorithm has been tested on the T0 and the T90 axis-symmetric models and showed an improvement compared to the classic NR. The details of the test are presented in Appendix D.

## Optimization of the initial shape

It is known that the NR method is very sensitive to the choice of the initial position and only provides a quadratic convergence if the initial position is "close enough" to the solution. In this context both stability and speed can be improved by optimizing the initial position. For the models considered in this report there is very little information on the generation of an initial cod-end shape for the axis-symmetric model and none for the 3D model.

### Axis-symmetric model

For the axis-symmetric cod-end model it is suggested [7] to start from a position of a cylinder with the cod-end entrance radius  $r_0$ . The nodes are offset from each other by the mean value between a knot and a twine length. Although, it is possible to converge from this initial position with an appropriate value of additional diagonal stiffness  $\alpha$ , it lacks the desired stability improved in the new initial position that is shown in Figure 3.9, p. 30.

In the new position, the relation between knot and twine length is preserved. It is ensured that the offset between the meridian nodes alternate between a slightly stretched twine or a knot length. This puts twines and knots under tension in order to use their full stiffness and thus the entries in the Jacobian matrix of a similar order of magnitude. In addition, in the end of the cod-end the shape transitions smoothly from  $r = r_0$  to  $r = 0$ . Influence of the new smoother initial shape on the number of iterations and the dependency on added stiffness is presented in Appendix D, and showed that this solution has a strong positive effect on the stability and speed of the axis-symmetric cod-end model.

**Algorithm 1:** Modified Newton-Raphson algorithm

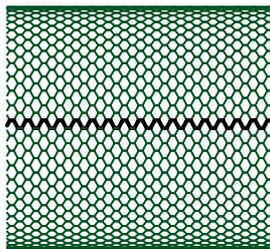
---

```

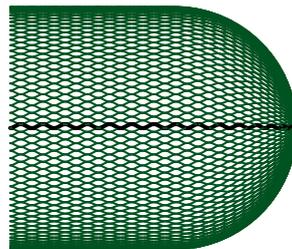
begin
   $i \leftarrow 0$ ;
   $tol_{residual}, tol_{disp}, \beta, R_{max} \leftarrow \text{SolverSettings}$ ;
   $found, correct, diverged \leftarrow false$ ;
  while not found and  $i < i_{max}$  do
     $\alpha, tol_{\alpha}$  from SolverSettings;
     $\mathbf{X}_i, |\mathbf{F}(\mathbf{X}_i)| \leftarrow \text{SolverSettings}$  initial shape;
    while  $|\mathbf{F}(\mathbf{X}_i)| > tol_{residual}$  or  $|\mathbf{h}_i| > tol_{disp}$  do
      Solve  $(\mathbf{J}(\mathbf{X}_i) + \alpha\mathbf{I})\mathbf{h}_i = -\mathbf{F}(\mathbf{X}_i)$ ;
      Find  $\lambda$  with line-search to satisfy  $|\mathbf{F}(\mathbf{X}_i + \lambda\mathbf{h}_i)| < (1 - \beta)|\mathbf{F}(\mathbf{X}_i)|$ ;
       $\mathbf{X}_{i+1} = \mathbf{X}_i + \lambda\mathbf{h}_i$ ;
       $i = i + 1$ ;
      if  $|\mathbf{F}(\mathbf{X}_i)| > R_{max}$  then
         $diverged = true$ ;
        break;
      end
      if  $|\mathbf{F}(\mathbf{X}_i)| < tol_{\alpha}$  then
        reduce  $\alpha$  and  $tol_{\alpha}$ ;
      end
    end
     $correct = \text{CheckCorrectSolution}(\mathbf{X}_i)$ ;
    if not diverged and correct then
       $found = true$ ;
    else
      increase  $\alpha$ ;
       $diverged = false$ ;
    end
  end
end

```

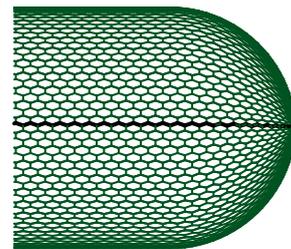
---



Suggested T0 tip



New T0 tip



New T90 tip

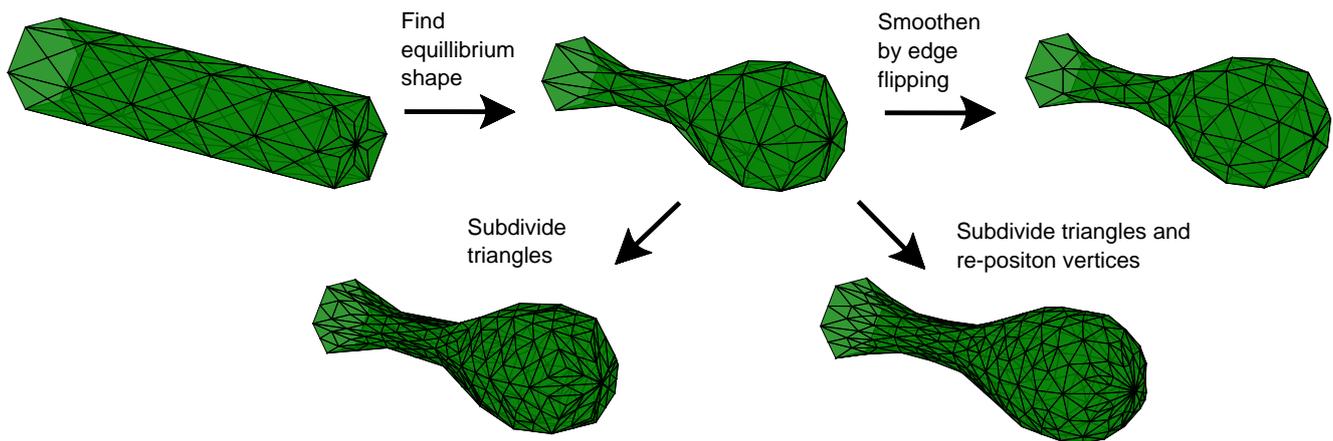
**Figure 3.9:** Comparison between the ending of a suggested and new cylindrical initial shape.**3D model**

For the 3D model, two main initial shapes are considered. The difference between them is whether the meshed panels are mapped to a cylinder with entrance radius (further referred to as *Shape 1*) or a cod-end shape pre-calculated with the axis-symmetric model (further referred to as *Shape 2*). *Shape 2* has an obvious advantage because it allows to start the iteration closer to the equilibrium shape of the cod-end, which is especially valuable for the calculations with fine mesh resolution.

In order to make the cylindrical shape more competitive, it is suggested to combine it with the mesh refinement and smoothing techniques. The idea is to start with a relatively coarse model and find its

---

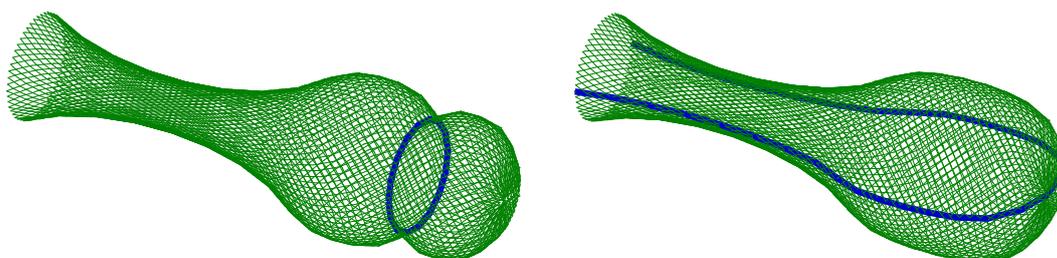
equilibrium shape. Thereafter, the triangles of the coarse model are subdivided (or subdivided and smoothed), which in turn becomes an initial shape for the next calculation with finer mesh. Two refinement methods and one smoothing method are considered. In the first refinement method each triangle can be subdivided into 4 smaller triangles by adding a vertex to each of its edges. The method is further referred to as *Shape 1a* or a plain subdivision. In the second refinement method the coordinates of new vertices are recomputed to preserve curvature continuity of the mesh within the neighbourhood of those vertices. The method is further referred to as *Shape 1b* or Loop subdivision [32]. Smoothing is performed with mesh edge-flipping method. The method attempts a trial flip of an edge shared by two neighbouring triangles. The flip is accepted if the average triangle skewness of the pair has been reduced. Refinement and smoothing methods are exemplified in Figure 3.10, p. 31 and described in more details in Appendix E.



**Figure 3.10:** Series of refinements and equilibrium solutions for a simple cod-end starting from a very coarse initial shape.

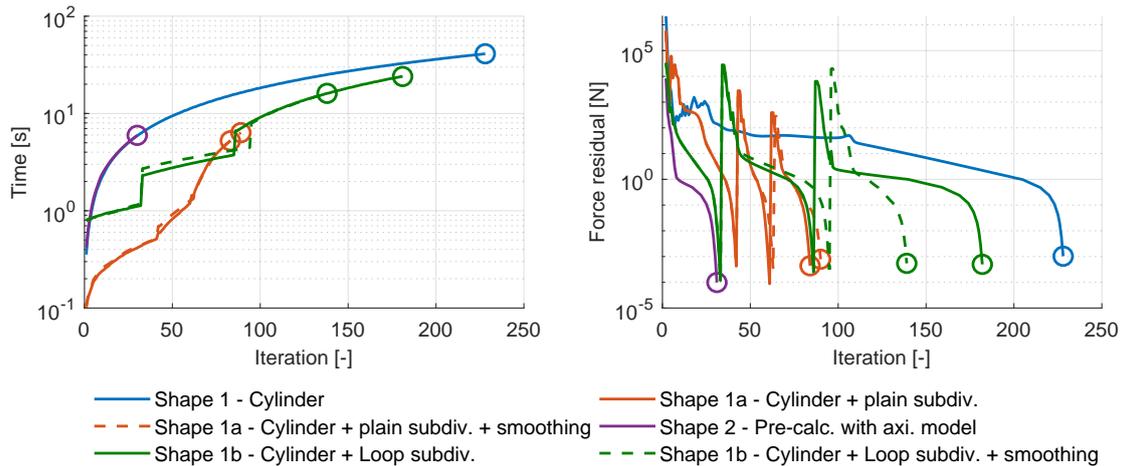
The amount of d.o.f in the refined model grows by factor of 4 since each triangle is subdivided into 4 smaller triangles. The refinement of the cylinder introduces new control parameters, such as number of subdivisions and number of restarts that influence the result. Subdivision and smoothing algorithms rely on mesh data structure containing connectivity and neighbourhood information for the vertices, triangles and edges of the triangular mesh. The data structure was not implemented in the current version of the C# program. Instead the potential of those methods was tested in MATLAB as prototypes.

The performance of the initial shape generation methods is compared based on 3 tests of a T0 diamond cod-end model with 50 meshes along, 50 meshes around, 100mm mesh side and 25 rows of meshes blocked by catch. Apart from the simple cod-end, there are tests for a cod-end with a round strap and a cod-end made of 2 panels joint with 2 selvedges. The equilibrium shape for those models is shown in Figure 3.11, p. 31. The round strap in test 2 is positioned at  $\frac{3}{4}$  of the total cod-end length and forms a ring with entrance radius. The selvedges in test 3 have length equal 90% of the fully stretched cod-end. For all 3 tests, the additional diagonal stiffness  $\alpha$  was fixed to the same value for all initial shape generation methods.



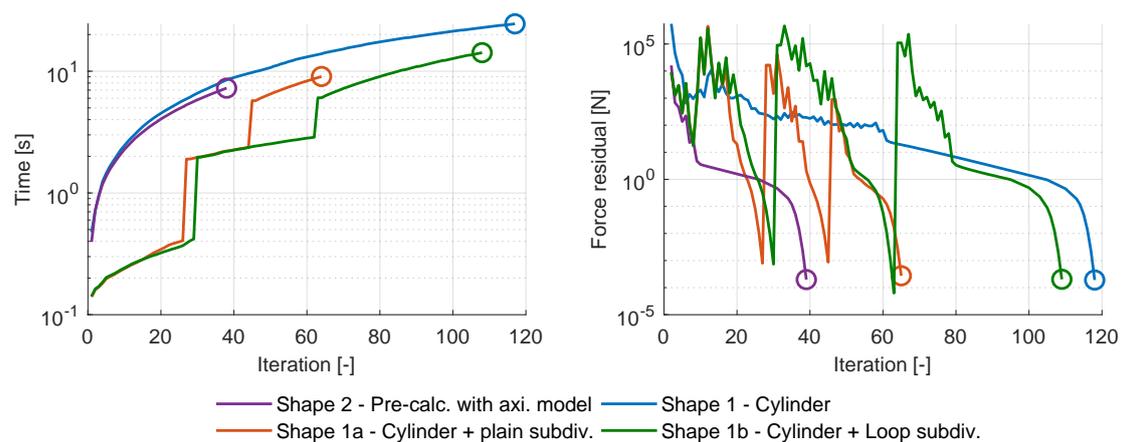
**Figure 3.11:** Equilibrium shapes for test 2 with a round strap and test 3 with two selvedges.

Figure 3.12, p. 32 shows the performance comparison of the initial shape generation methods, both in terms of time and the amount of iterations. The test is performed on the simple cod-end with no attachments. The model contained  $\approx 1500$  d.o.f. The figure shows that if the mesh resolution is fine from the beginning, the cylindrical initial shape converges slowly, since it is quite far from the equilibrium. The refinement technique *Shape 1a* greatly improves the cylindrical initial shape, since the original poor cylindrical initial shape is only used with a very coarse mesh resolution. For this case, it converges in the same time as the *Shape 2*. Loop subdivision method in *Shape 1b* did not improve the plain subdivision as it was expected. Edge flipping method moderately improves the performance of the Loop subdivision method and affects the plain subdivision method only slightly. For two next tests it was decided not to include smoothing, in order not to alter the triangles in the proximity of a strap or selvages.



**Figure 3.12:** Convergence speed test with 6 different initial shape approaches for the simple cod-end model.

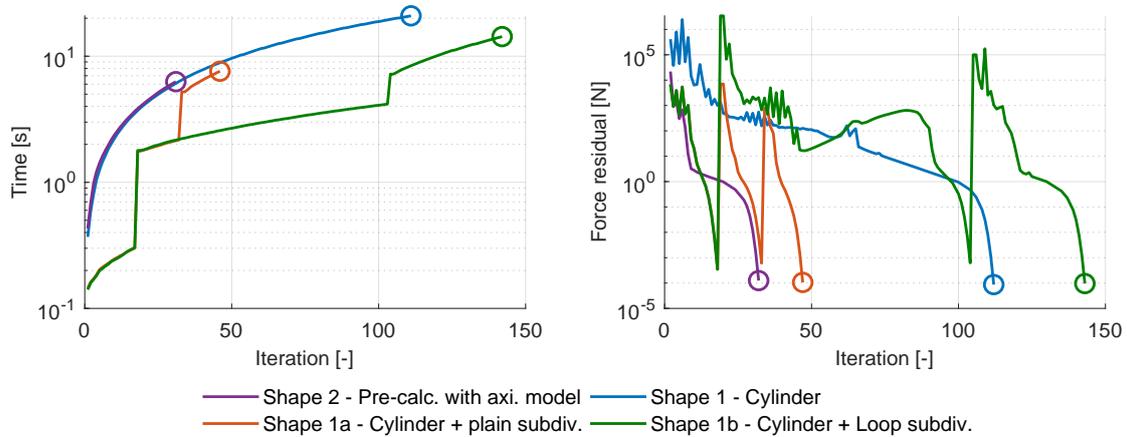
Figure 3.13, p. 32 shows the test performed on the cod-end with the round strap. *Shape 2* converged faster, steadier and in fewer iterations than other shapes. *Shape 1a* again showed great results being only slightly slower than the *Shape 2*. However, when *Shape 1a* reached second refinement stage, there were some fluctuations in the force residual on its convergence path to the solution. *Shape 1b* is still being faster than *Shape 1*, however, in this test the difference between them is smaller. *Shape 1* outperforms *Shape 1b* in terms of stability, due to the fact that it converges to the solution in a more steady fashion.



**Figure 3.13:** Convergence speed test with 4 different initial shape approaches for the cod-end model with a round strap

Figure 3.14, p. 33 shows the test performed on the cod-end made of 2 panels joined with selvages. The situation is similar to first 2 tests. *Shape 2* showed the same steady and fast convergence rate. *Shape 1a* is being slightly slower, but more stable compared to the test 2. *Shapes 1* and *Shape 1b* converged with many fluctuations in the force residual. It is observed that the *Shape 1b* starts with the biggest residual on restart,

which suggests that the re-positioning of vertices yields the new cod-end shape that is farther away from the equilibrium shape.



**Figure 3.14:** Convergence speed test with 4 different initial shape approaches for the cod-end model with two selvages.

The tests show that an initial shape based on a pre-calculated equilibrium position of the axis-symmetric model is an excellent choice. The pre-calculation of the shape takes milliseconds. The solution with this initial shape converges in the least amount of iterations, and shows better stability compared to the other approaches. Plain subdivision method is a great alternative with a good potential. The downsides of this approach compared to the *Shape 2*, is the existence of extra parameters like the amount of refinement steps and the amount of subdivisions per refinement. The parameters can both improve and degrade the performance of the method. Furthermore, the sub solutions of the coarser models might need different  $\alpha$  for the optimal convergence.

With some improvements, smoothing can potentially improve the subdivision methods. The downsides are, that edge flipping cannot be performed on the edges where the panel joins itself. Also, edge flipping along the edges that accommodate a round strap or a selvedge, can lead to the incorrect geometry where bar elements are not co-planar to the triangular elements.

Loop subdivision scheme showed less potential than the plain subdivision scheme, probably due the fact that when the new Cartesian vertex coordinates are generated and re-positioned they do not correspond well with new  $uv$  mesh coordinates. The new method of preserving this consistency between the Cartesian and  $uv$  mesh coordinates might improve the efficiency of this method.

It should be noted, that this comparison is meant as a qualitative assessment of the highlighted methods. Reader should not pay attention to the actual time elapsed, but rather to the influence of different initial shapes to the performance since calculations were done in a prototype MATLAB script, that was intentionally not optimized for speed.



# Model comparison

In this chapter the models described earlier are compared. The goal is to identify the cases where the models converge to the same results, and the cases when the results are different. In the first scenario it would mean that here the axis-symmetric model is preferred due to its speed. In the second scenario, one have to run a heavier 3D model, in order to take all the mechanical behaviour into account. An estimation on how every combination of the input parameters influences the comparison results requires an extensive study, which is not the purpose of this project. Therefore, the approach is to limit the comparison to the set of parameters that are believed to influence the results the most. The comparison is conducted with following limitations:

- Only T0 codends are considered. There are 3 basic test cod-ends, whose specifications are given in Table 4.1, p. 35. Modifications to the basic test cod-ends such as change in  $EI$  are described in relevant subsections dedicated to specific tests.
- Weight is negligible compared to other loads, therefore is not taken into account in any of the models.
- Test for cod-ends are run with the sequence of catches from 10 to 50 blocked meshes with the step of 5.
- Only one towing speed of  $0.8 \text{ ms}^{-1}$  is considered.
- Five comparison parameters are chosen: maximum radius, maximum length, catch volume, resultant drag force and a mean mesh opening angle.

**Table 4.1:** Basic test cod-end specifications.  $M$ . stands for *mesh*.

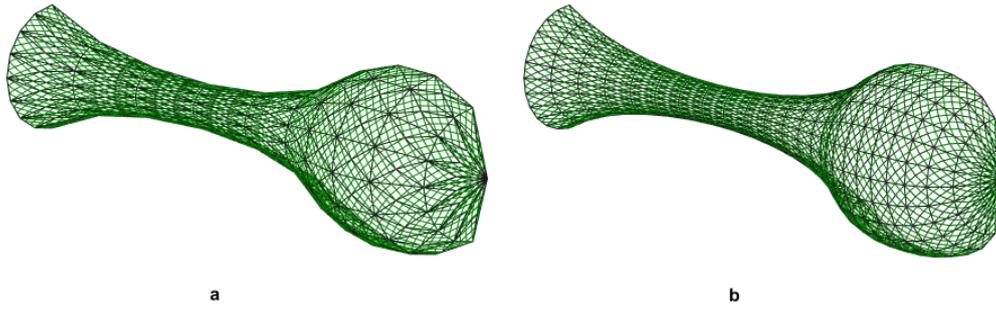
Cod-end	Ent. radius	M. along	M. around	M. side	Twine thickness	EA	EI
1	0.4 m	50	50	80 mm	single 4 mm	1000 N	0 Nmm <sup>2</sup>
2	0.4 m	50	80	100 mm	single 6 mm	1000 N	0 Nmm <sup>2</sup>
3	0.4 m	50	100	120 mm	single 8 mm	1000 N	0 Nmm <sup>2</sup>

## 4.1 Mesh convergence study

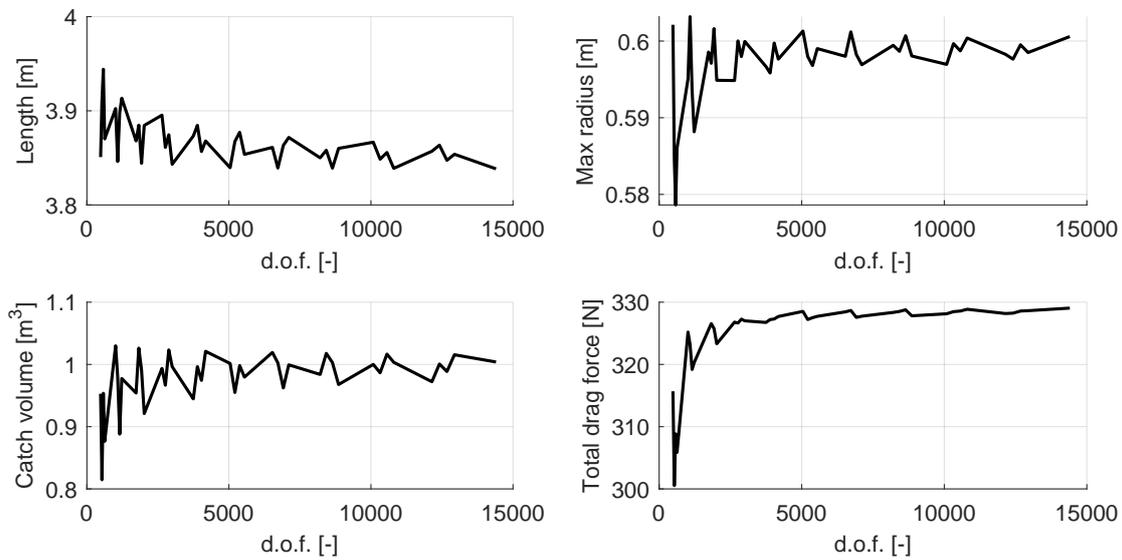
The results of the numerical model based on the 3D finite element method depend on the level of discretization, or in other words size and quality of the triangular mesh. The effect of the meshing on a cod-end model is exemplified in Figure 4.1, p. 36. In order to provide a reliable comparison the mesh size is chosen based on the mesh convergence study.

For the cod-end it is possible to manipulate the mesh by changing the number of elements along or number of elements across or both at the same time. It is considered ineffective to only increase the discretization at one of the mentioned directions since it has a bad influence on the skewness of the elements. Therefore, for the convergence study the discretization is undertaken both along and across the cod-end. Figure 4.2, p. 36 shows how the result parameters such as maximum radius, maximum length, catch thickness and resulting drag force approach their asymptotic value as the mesh is refined.

It is noted, that reaching approximately  $\approx 5000$  d.o.f. all comparison parameters begin to oscillate around their corresponding asymptotic values. Considering the magnitude of the y-axes the oscillations are deemed negligible, therefore the mesh with  $\geq 5000$  d.o.f. achieved by evenly discretizing the cod-end along and around are considered well suited for the comparison. For the cod-ends 2 and 3 from Table 4.1, p. 35 a separate mesh convergence is undertaken and suggested similar results.



**Figure 4.1:** Coarse and fine discretization cases of the same cod-end. Case **a**: a triangle covers 5 netting meshes along and 3 netting meshes across. Case **b**: a triangle covers 2 netting meshes along and 2 netting meshes across.



**Figure 4.2:** Mesh convergence study with 4 parameters for the basic cod-end 1 with 20 meshes blocked by catch.

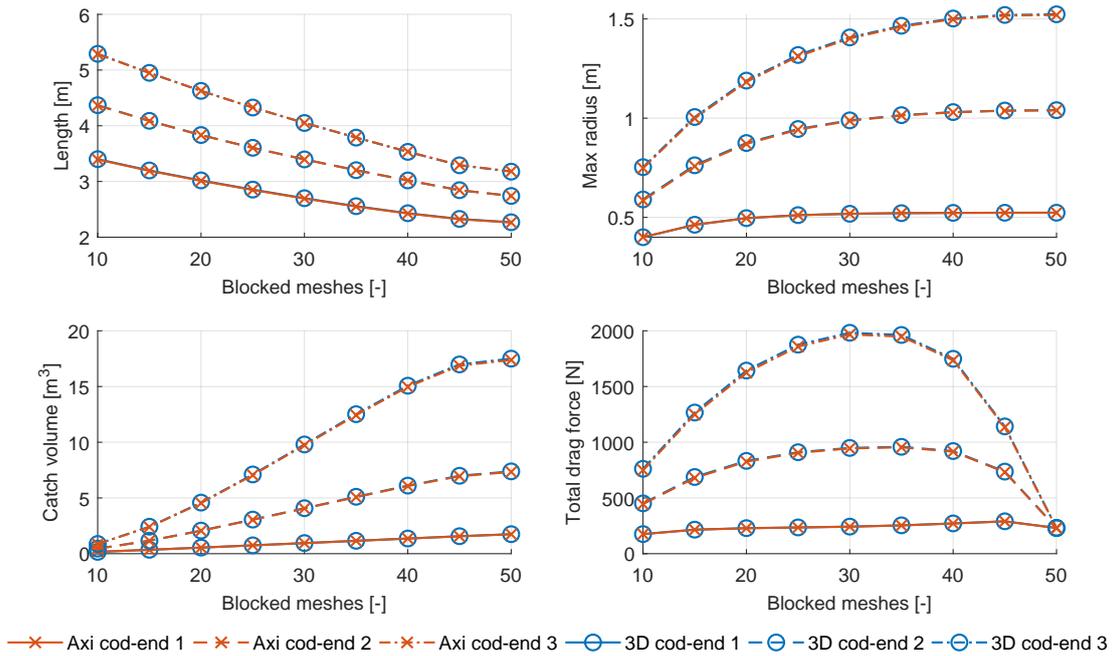
## 4.2 Comparison 1 - basic test

In the basic test a number of parameters is neglected to leave only the mechanical behaviors that both models can take into account. Those neglected parameters are knot size, drag on the netting, bending and opening resistance. Since in the present axis-symmetric model, the knot size cannot be ignored completely, it is set to 1/1000 of the twine size. In this scenario is expected for the models to show relatively similar results. The comparison is presented in Figure 4.3, p. 37 and Figure 4.4, p. 37.

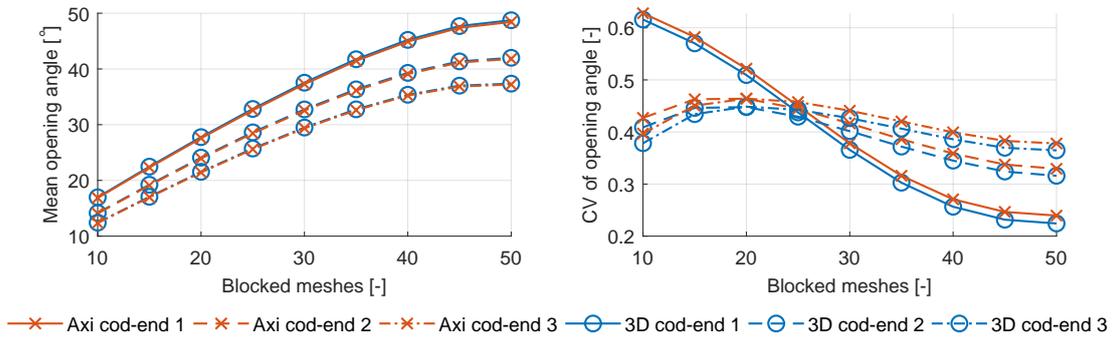
It is shown that two models correspond well in case of all four comparison parameters. Notably, the total drag for the full cod-end matches for all 3 models. Such behaviour is justified in, since according to [22] the total drag can be approximated as the the maximum pressure applied on frontal area of the catch. In case of the fully blocked cod-end the catch area is the entrance area that is equal for all 3 cod-ends. Both models predict the same mesh opening angle, however the variation of the angle is slightly greater for the axis-symmetric model. It could be due to the difference in the discretization of two models.

## 4.3 Comparison 2 - Effect of the knot size

As mentioned before, the "knotless" version of the axis-symmetric model corresponds to knot to twine length ratio of 1/1000. In this test, the basic test is modified and other bigger ratios of 1/100 1/50 1/10 are considered. Drag on the netting and the bending stiffness are still neglected. The effects of different knot to twine ratios are presented for 3 cod-end models in Figure 4.5, p. 38 and Figure 4.6, p. 38.



**Figure 4.3:** Comparison 1. Between axis-symmetric and 3D models for 3 cod-ends from Table 4.1, p. 35. Knot size, drag on the netting, and bending stiffness are not considered. The discretization for 3D model is  $\approx 10000$  d.o.f.



**Figure 4.4:** Comparison 1. Mean mesh opening angle and the coefficient of variation for the angle for the basic test.

It is seen that for the same knot to twine ratio the difference between the models scales with the amount of meshes in the cod-end. For the cod-end model 1 this effect is mostly represented in increasing the total length of the cod-end. However, the effect is more significant for the cod-ends 2 and 3 with more meshes in the circumference. The error also varies depending on the amount of the catch in the cod-end. Table 4.2, p. 38 shows the mean error over all catches in estimating the mean mesh opening angles for 3 cod-ends.

Depending on the user imposed error tolerance, the ratios 1/100 and 1/50 might be considered acceptable, and the 3D model can still be used for those cod-end. In contrast, to accurately calculate the cod-ends with larger knot to twine ratios, 3D model needs to be update with another type of triangular element that is based on the hexagonal mesh representation [11].

## 4.4 Comparison 3 - effect of the drag on the netting

In the previous tests only the drag on the catch was considered. Here, the drag on the netting is included in the 3D model. As mentioned in Section 2.1, p. 7, netting drag forces are subjected to the most simplifications, because the flow field inside the cod-end is not yet fully understood. There are two sub tests to evaluate the

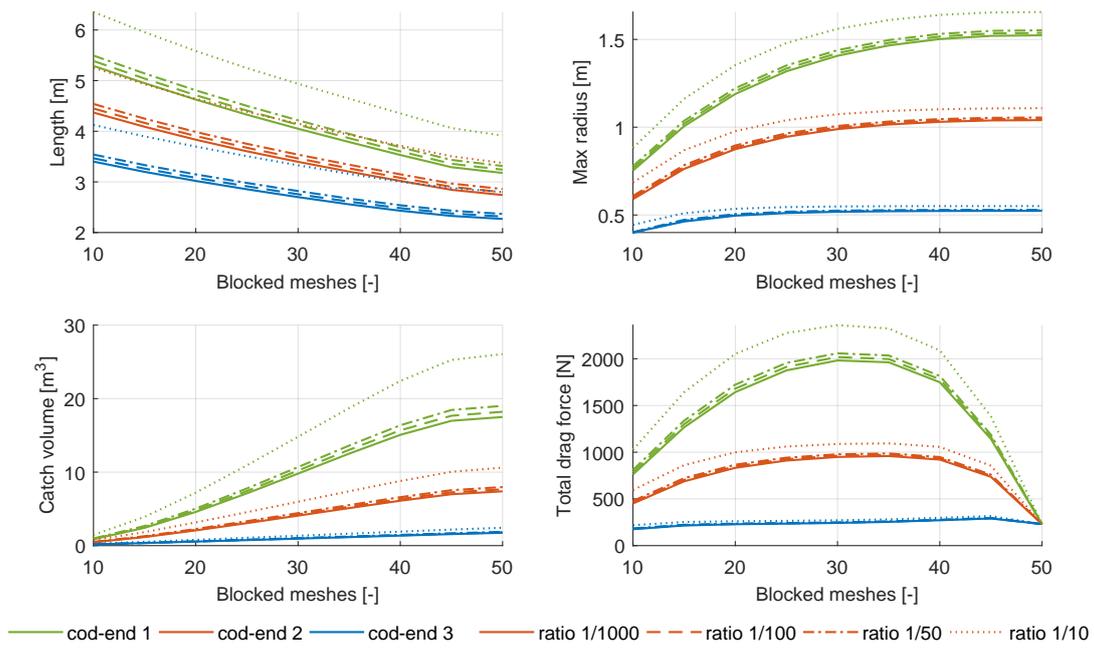


Figure 4.5: Comparison 2. 3D model to the axis-symmetric model with different knot to twine ratios.

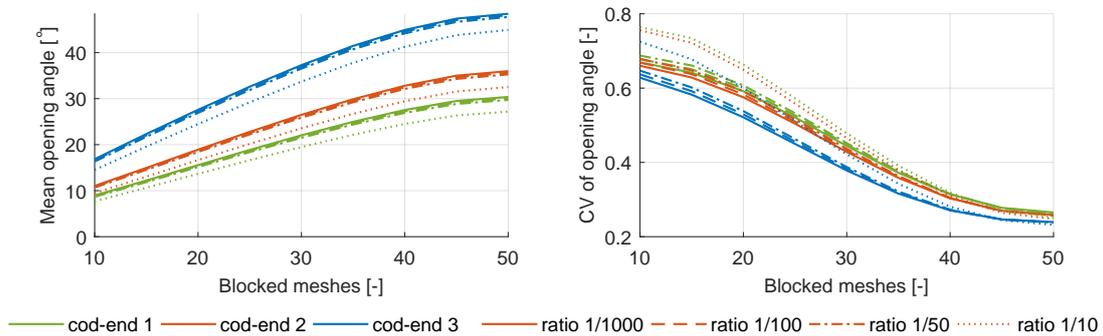


Figure 4.6: Comparison 2. 3D model to the axis-symmetric model with different knot to twine ratios.

Table 4.2: Comparison 2. Mean error over all catches in mean opening angle relatively to the 1/1000 knot to twine ratio.

Codend	1/100	1/50	1/10
1	-1.0 %	-2.1 %	-10.0 %
2	-1.1 %	-2.4 %	-11.3 %
3	-1.2 %	-2.5 %	-11.8 %

netting drag. In the first one, the netting drag is applied to the entire netting. In the second one, the drag is applied only to the netting that the flow can pass through, meaning not blocked by catch. In both cases no flow speed reduction is given to the twines that are in a shadow of the other twines, because the 3D model considers uniform and steady flow. The pressure and friction drag coefficients are calculated with Eq. (2.29) and Eq. (2.30) from Section 2.1, p. 7.

### 4.4.1 Netting drag applied to all twines

The comparison of the axis-symmetric model and the 3D model with the drag applied to all twines is presented in Figure 4.7, p. 39 and Figure 4.8, p. 39.

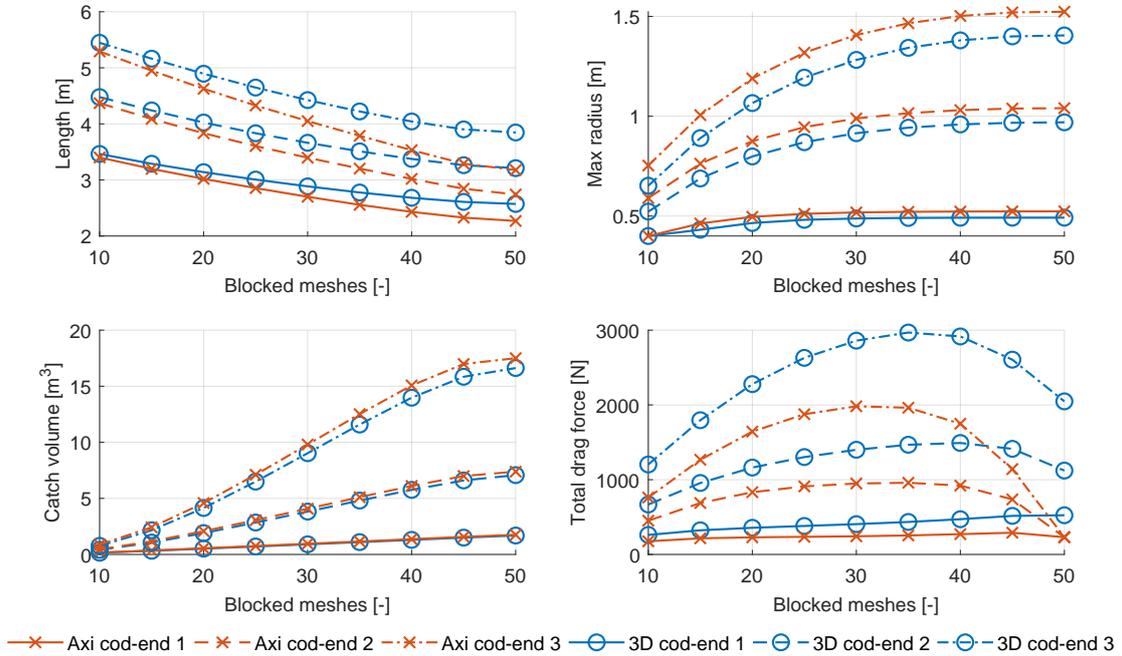


Figure 4.7: Comparison 3. Axis-symmetric model to the 3D model where drag on the netting is included for all twines.

Due to the extra drag on the netting the length of the cod-end increases and the radius decreases which also leads to the decrease in volume. The effects are more distinct in case of the cod-end models 2 and 3 that have a bigger twine thickness. The error varies depending on the amount of catch in the cod-end. Table 4.3, p. 40 shows the mean error in estimating the comparison parameters for the 3 cod-ends. This case probably leads to an overestimation of the drag forces on the netting, especially for the cod-ends with bigger total number of twines for the drag to take place. In contrast to the basic test, the drag forces for 3 models do not meet in one value when the cod-end is full. This in turn produces the biggest error in the drag force comparison, especially for model 3 with the biggest twine thickness. However, at that stage all the meshes are blocked by catch, so the value of the drag is highly overestimated. The effect of increasing the cod-end length and decreasing its radius influences mesh opening angle. The meshes tend to be more closed, and the opening angle variation is slightly lower than in the axis-symmetric model.

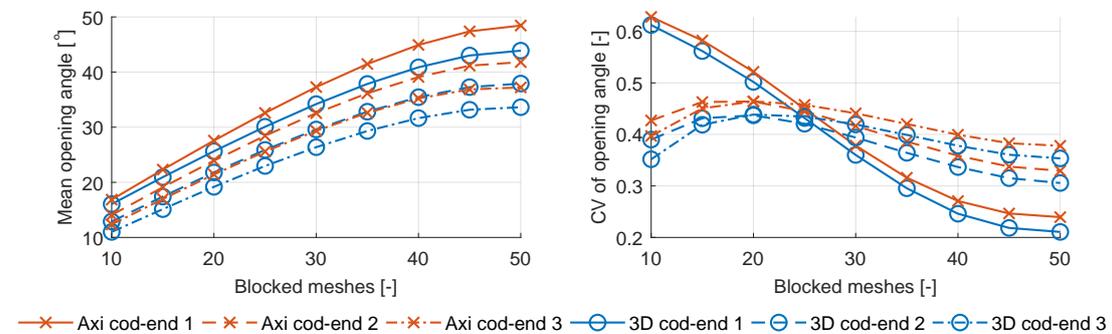


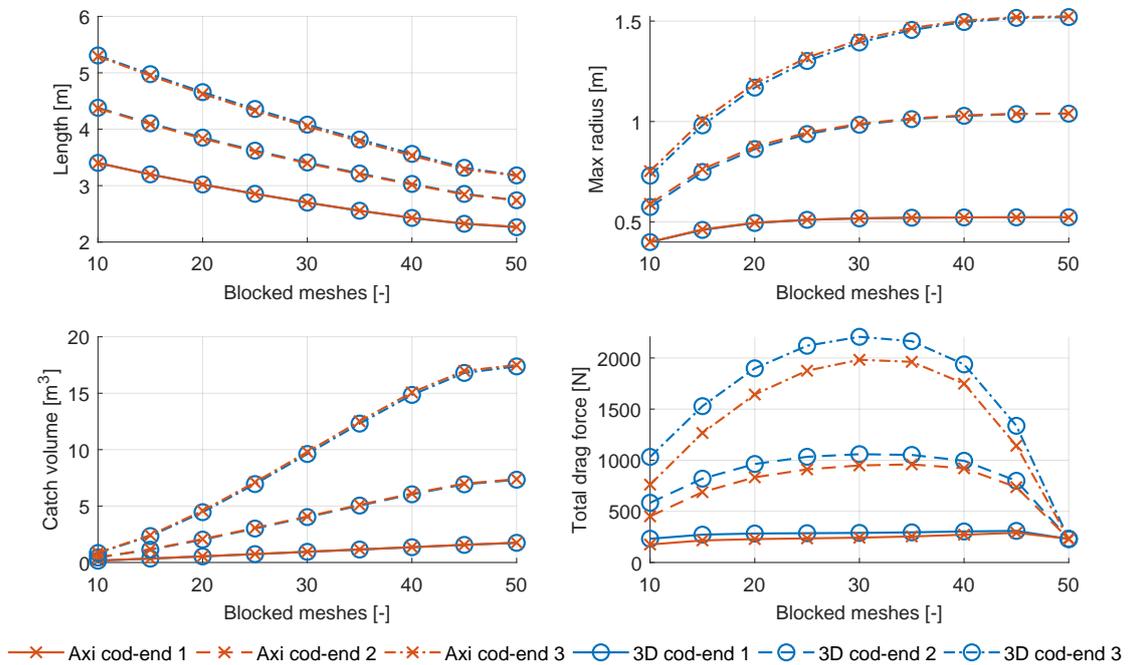
Figure 4.8: Comparison 3. Axis-symmetric model to the 3D model where drag on the netting is included for all twines.

**Table 4.3:** Comparison 3. Mean error of the 3d model with netting drag for all twines relative to the axis-symmetric model.

Codend	Length	Radius	Volume	Total drag	Mean opening angle
1	7.2 %	-5.4 %	-5.3 %	70.2 %	7.8 %
2	8.7 %	-8.1 %	-7.0 %	90.0 %	9.1 %
3	10.6 %	-9.5 %	-8.4 %	139.6 %	10.3 %

#### 4.4.2 Netting drag applied to twines not in contact with the catch

The comparison of the axis-symmetric model and the 3D model with the drag applied only to the twines not in contact with catch is presented in Figure 4.9, p. 40 and Figure 4.10, p. 41.

**Figure 4.9:** Comparison 3. Axis-symmetric model to the 3D model where drag on the netting is included for twines not in contact with the catch.

Similarly to the previous test, the length of the cod-end increases and the radius decreases, but now much less, even for the cod-end 3. Addition to the total drag due to contribution of the drag on the netting is observed, but as the cod-ends accumulate catch, this addition approaches zero. Table 4.4, p. 40 shows the mean error in estimating the comparison parameters for the 3 cod-ends.

**Table 4.4:** Comparison 3. Mean error of the 3d model with netting drag not in contact with the catch relative to the axis-symmetric model.

Codend	Length	Radius	Volume	Total drag	Mean opening angle
1	0.0 %	-0.1 %	-1.1 %	17.4 %	0.2 %
2	0.3 %	-0.9 %	-1.7 %	12.8 %	-1.0 %
3	0.6 %	-1.2 %	-2.1 %	14.9 %	-1.8 %

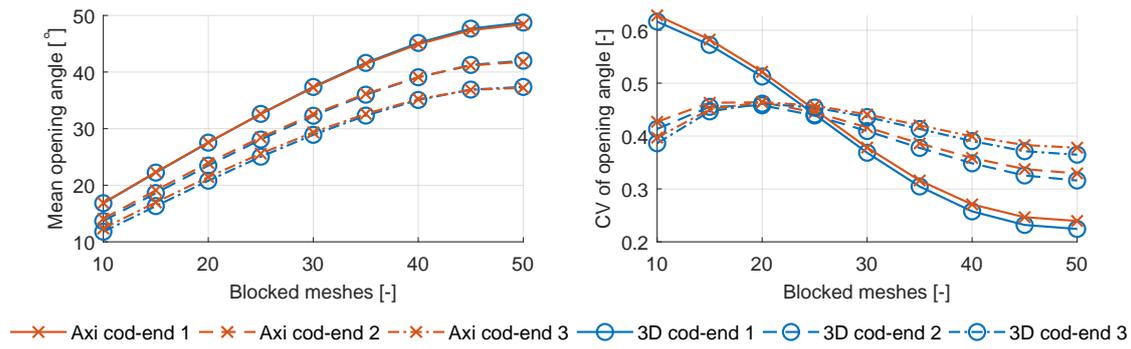


Figure 4.10: Comparison 3. Axis-symmetric model to the 3D model where drag on the netting is included for twines not in contact with the catch.

### 4.5 Comparison 4 - effect of mesh opening

Previous test did not take into account the bending stiffness of the netting. In the T0 configuration the mesh resistance to opening helps to close the meshes. Those effects are taken into account in the 3D model and compared to the axis-symmetric model. The 3D model only takes constant values of the  $EA$  and  $EI$  into account, which is a simplification. In reality those stiffness values are non-linear and depend on the current opening of the mesh and the twine tension. Therefore, it is deemed unreasonable to look for a precise definition of the  $EA$  and  $EI$  and the maximum stiffness values were used as if the twine is a cylinder with the diameter  $d$  given in Table 4.1, p. 35. The mesh opening stiffness  $H$  can be represented through the bending stiffness  $EI$  and mesh side  $M$  as shown in Eq. (4.1) [33]. The elasticity modulus for the PE netting is taken as  $1000\text{Nmm}^{-2}$ . The comparison is presented in Figure 4.11, p. 41 and Figure 4.12, p. 42.

$$H = \frac{12EI}{M} \tag{4.1}$$

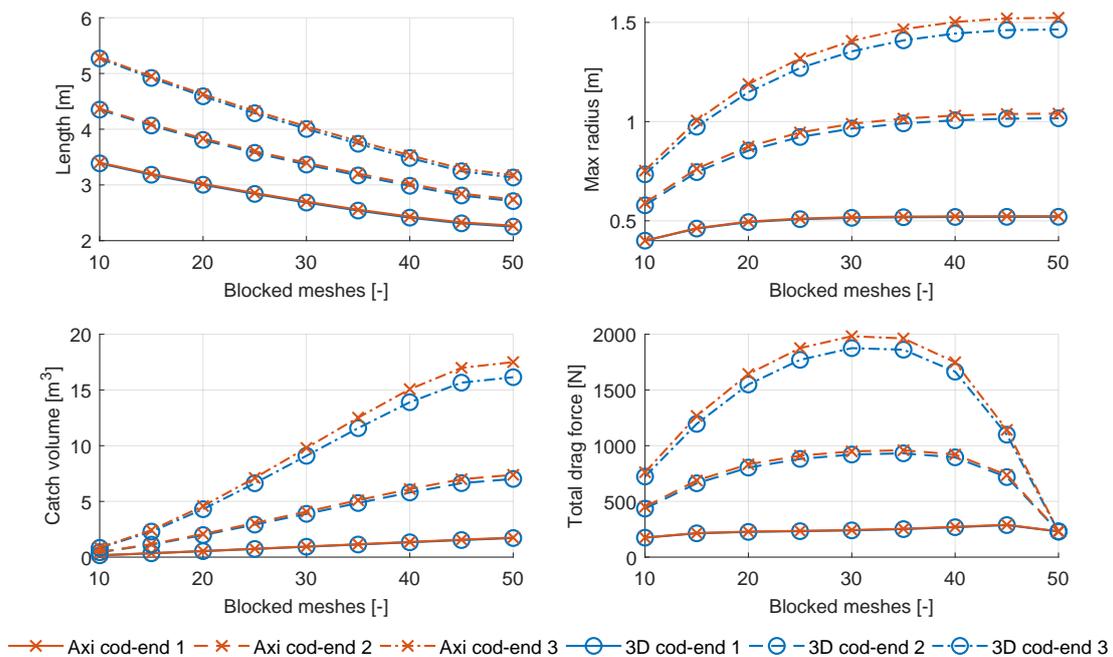
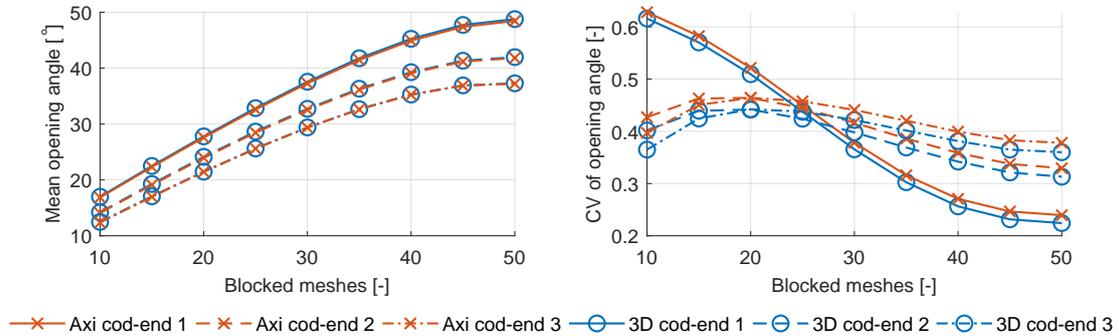


Figure 4.11: Comparison 4. Axis-symmetric and 3D where opening stiffness is taken into account.

It is seen that this mechanical effect does not affect the comparison parameter significantly especially for the cod-end 1 with the lowest twine thickness and bending stiffness (See Table 4.5, p. 42). The deviation from the axis-symmetric model is seen mostly in reduction of the radius and the catch volume. Remarkably, the mean opening angle almost have not changed. It could be due to the fact that, together with the decrease in the radius there is also a decrease in the length that cancel out in the total contribution to the mesh opening angle.



**Figure 4.12:** Comparison 4. Axis-symmetric and 3D where opening stiffness is taken into account.

**Table 4.5:** Comparison 4. Mean error of the 3d model with opening resistance to the axis-symmetric model.

Codend	Length	Radius	Volume	Total drag	Mean opening angle
1	-0.6 %	-0.5 %	-2.0 %	-1.0 %	0.7 %
2	-0.9 %	-2.3 %	-4.6 %	-2.8 %	0.6 %
3	-1.0 %	-3.6 %	-6.7 %	-4.6 %	0.3 %

## 4.6 Discussion

The comparison of the axis-symmetric and 3D models is presented focusing on few chosen parameters. It is seen that the axis-symmetric model can be adjusted to show the similar results in the cases where the drag on the twines is small. Nevertheless, even with the twine drag included, the geometrical result parameters can be estimated within 10 % error. This fact also explains, why it is so efficient as an initial shape for the 3D FEM model.

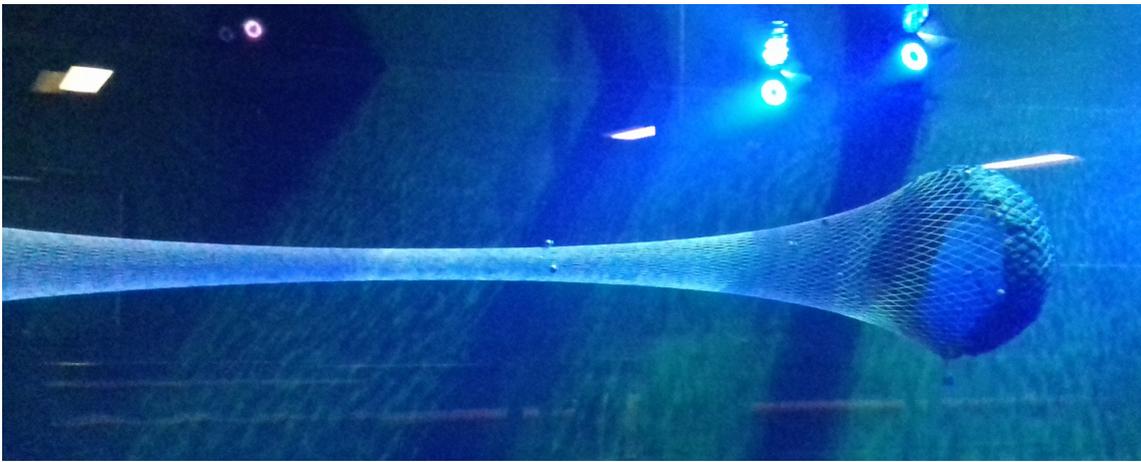
---

# Validation

---

## 5.1 Shape estimation of a single panel cod-end

In this chapter the models are validated against the physical tests carried out in the flume tank at SINTEFF OCEAN based in Hirtshals Denmark. The tank specification are found in Appendix F.



**Figure 5.1:** Shape of a single panel cod-end with 8 rows of meshes blocked is being measured in the flume tank.

A cod-end made of a single panel can be assumed to be axis-symmetric, therefore it can be used to validate both axis-symmetric and 3D numerical models. In practice, the use of cod-ends and extension pieces which are made of only one piece of net material and have only one selvedge is prohibited [34], therefore the cod-end was specially manufactured for the experiment. The specifications are presented in Table 5.1, p. 43.

**Table 5.1:** Single panel cod-end specifications.

Parameter	Value
Type	T0
Meshes along	50
Meshes around	48
Bar length	50 mm
Twine thickness	Single 1.4 mm
Knot length	1.4 mm

### 5.1.1 Setup

In the flume tank the cod-end model was attached to a metal ring. The metal ring was held by four ropes emerging from the metal loops left on the top and the bottom of the opposite walls of the tank as shown in Figure 5.2, p. 44. In this way the structure was supported by four mooring lines pre-tensioned by hand.

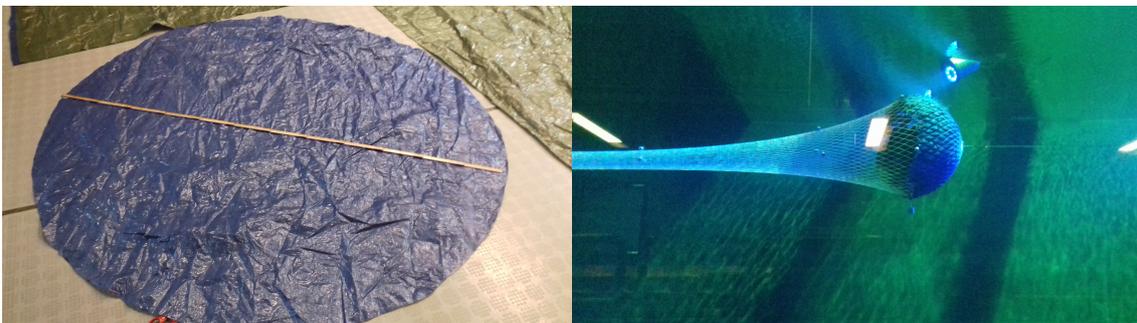
Tests with 8 and 5 blocked meshes were carried out with water speed of  $0.4\text{ m s}^{-1}$  and  $0.8\text{ m s}^{-1}$  each. Blocking was established by fixing a circular patch of impermeable fabric inside the cod-end at the specific

---



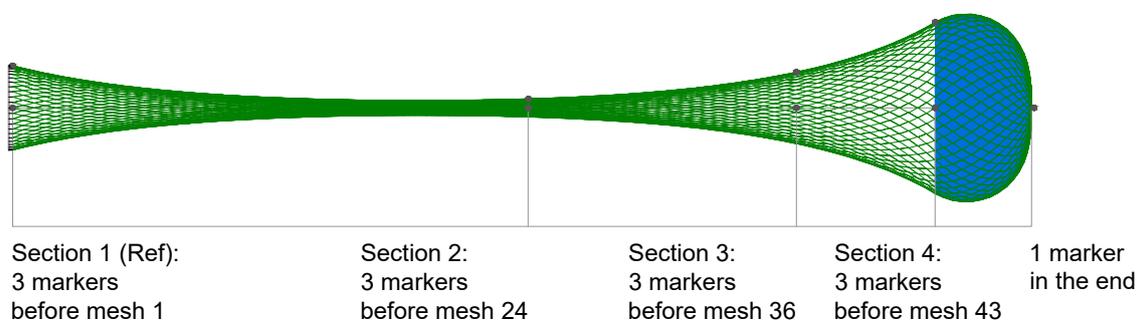
**Figure 5.2:** Entrance ring with the cod-end and markers attached (left) and the ring on the mooring lines in the flume tank (right).

row of meshes as shown in Figure 5.3, p. 44. Small number of blocked meshes was chosen for the easiness of installing the patch as well as because wrapping a circular patch to semi-spherical end part of the cod-end will give less excess patch material as opposed to wrapping it to a much longer "bullet-shape" that naturally appears if the cod-end has a relatively large catch.



**Figure 5.3:** Circular patch of impermeable material cut (left) and fixed in the cod-end to block 8 rows of meshes (right).

Four control sections including a reference section were chosen for comparison with the numerical models as shown in Figure 5.4, p. 44. A section is positioned at a certain amount of meshes away from the entrance and supplied by three markers. Three markers were used to fit a circle and estimate a radius of the control section. In addition the distance from the center of a control section to the reference section was calculated. Reference section passed through the very first mesh of the cod-end. The markers were positioned before meshes 1, 24, 36, catch blockage as well as the very end of the cod-end. The marker size was 28mm which is relatively large compared to the mesh size, and has to be accounted for in the data processing.



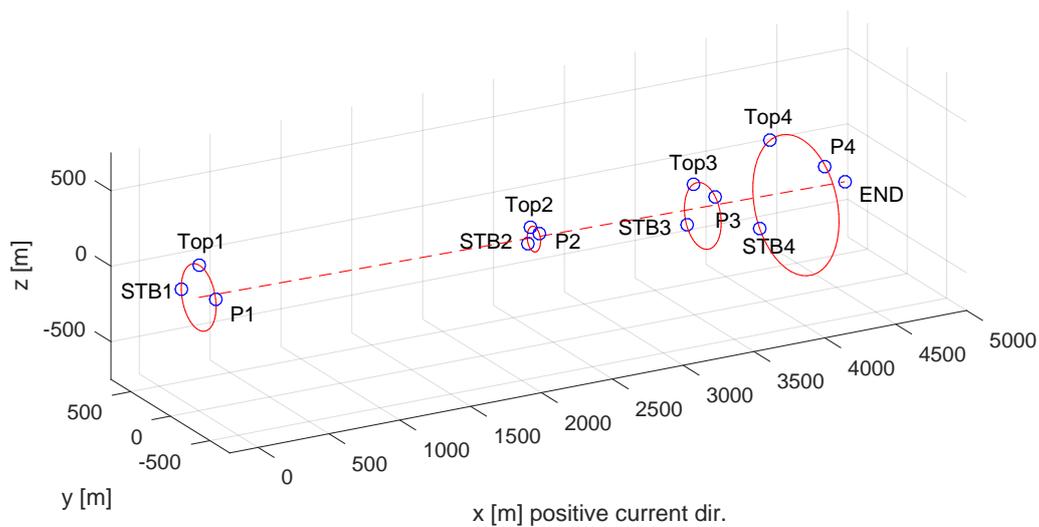
**Figure 5.4:** Marker positions for a test with 8 rows of meshes blocked by the catch.

The motion tracking system Qualisys [19] was used to sample the spatial coordinates of the markers. In total 1100 samples were taken, and the sampling time was 100s. Calibration was used to calculate the

positions of the cameras. There were two bridges with four underwater cameras each available for the experiment. The calibration of the system happened by lowering the test XY-frame with four markers and known dimensions down the flume tank, and sampling its position together with arbitrary movement of a calibration rod with two markers. After that, the position of each marker was possible to track, provided that it was visible by at least two cameras to perform a triangulation. Only the shape of the cod-end was measured during the test. The total drag on the cod-end was not measured.

### 5.1.2 Results

The cod-end placed in the flume tank under the uniform current is an unstable structure. During the test the single panel cod-end did not only oscillate under the effects of current, but also rotated around its axis, since there were no selvages to provide rotational stiffness. An example of the rotation can be seen in Figure 5.5, p. 45. The problem was partially solved by attaching a small weight under the cod-end Figure 5.3, p. 44. The weight was chosen so that it does not affect the cod-end's deformation and position significantly, but helps to suppress the rotation of the cod-end. Considering the given conditions, it cannot be expected to clearly see all the markers over the entire sampling time. For each marker the fill ratio can be calculated, that shows how many percent of the total time it was possible to see the marker and determine its spatial coordinates. In order to perform circle fitting to determine the radius and the center position of a control section all 3 markers must be present. The amount of time it was possible to see all three markers of a section at the same time can be represented by a fill ratio for a section as shown in Table 5.2, p. 45. It is seen that for the tests 3 and 4, the fill ratio is very low or zero in the most interesting rear sections, therefore it was decided to discard this data and continue only with tests 1 and 2.



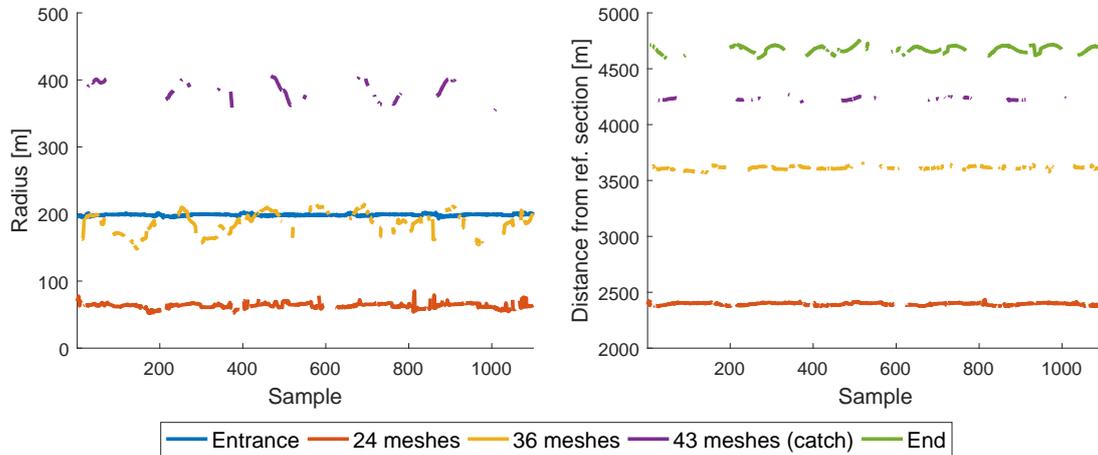
**Figure 5.5:** An example of a sample where all markers are visible. It can be noticed that the cod-end is moderately twisted around the axis. Markers in the control sections are numbered accordingly and their abbreviations stand for: STB - *starboard* or left marker, P - *port* or right marker, Top - top marker.

**Table 5.2:** Fill ratio for control sections

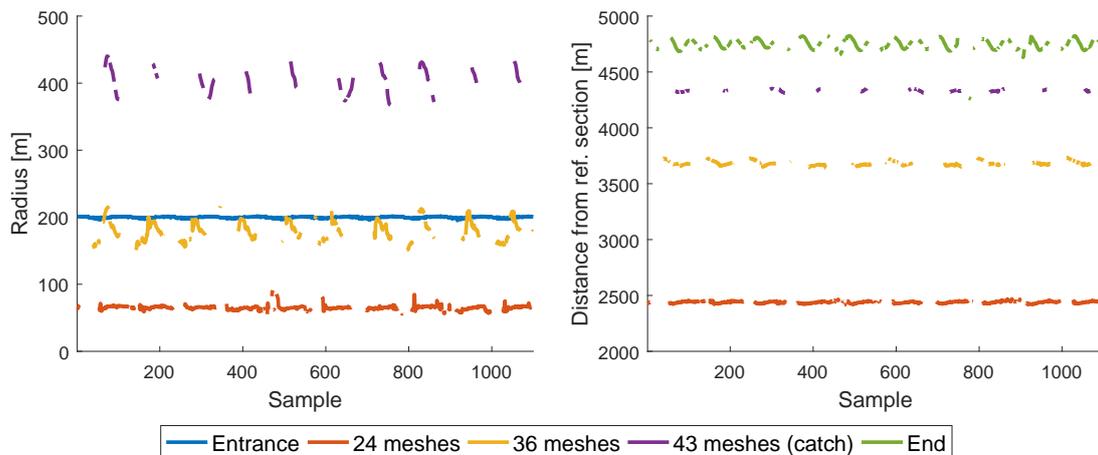
Test nr.	Water speed	Blocked meshes	Section 1(Ref)	Section 2	Section 3	Section 4
1	$0.4\text{m s}^{-1}$	8	99.6 %	90.9 %	54.9 %	18.7 %
2	$0.8\text{m s}^{-1}$	8	100 %	74.5 %	44.9 %	18.2 %
3	$0.4\text{m s}^{-1}$	5	95.7 %	36.2 %	1.8 %	0.0 %
4	$0.8\text{m s}^{-1}$	5	88.0 %	23.6 %	3.0 %	2.0 %

Raw data for the tests 1 and 2 is presented in Figure 5.6, p. 46 and Figure 5.7, p. 46. The fill ratio for the

data can be easily visualized. The effect of the instability of the cod-end can be seen in the fluctuations of the data. Marker size was accounted for in the calculation of the section radiuses as well as in the calculation of the end distance. It was done because Qualisys measures the position in the middle of the exposed area of the marker. In the best case when the entire marker is visible, its position is one marker radius away from the netting position that is measured.



**Figure 5.6:** Raw data for the test 1: Section radiuses (left) and distances from the reference section (right).

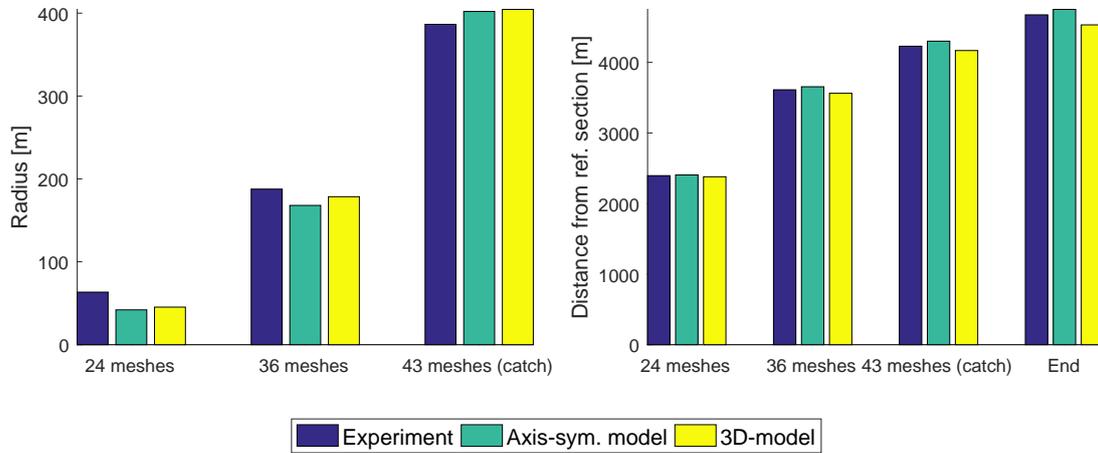


**Figure 5.7:** Raw data for the test 2: Section radiuses (left) and distances from the reference section (right).

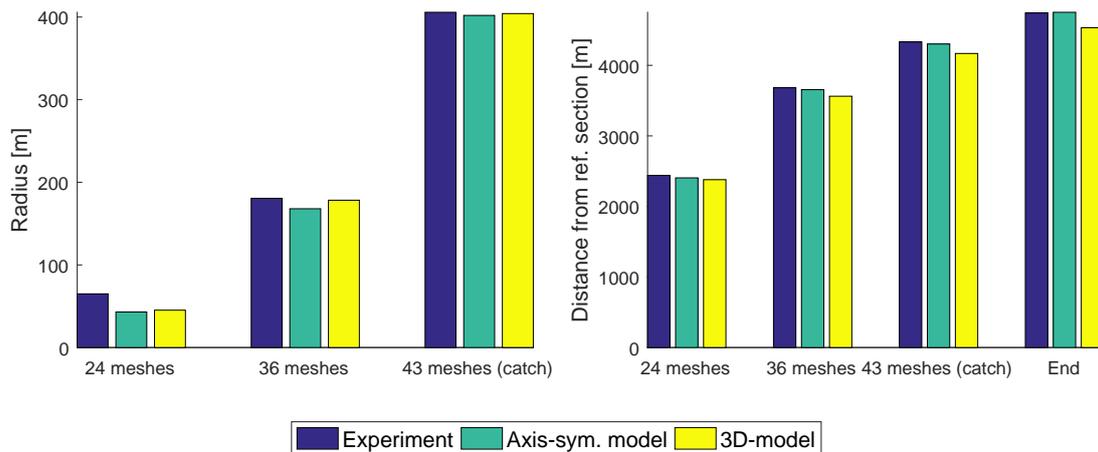
The cod-end specifications were input to the numerical models. The axis-symmetric model included the twine tension and catch pressure forces and accounts for the both twine and knot size in the shape calculation. The 3D model included twine tension, drag on the netting and catch pressure. Since the forces were not measured during the experiment, the pressure drag and friction drag coefficients for each element were calculated based on the expressions presented in Section 2.1, p. 7 Eq. (2.29) and Eq. (2.30).

Both twine bending outside the plane and mesh opening resistance were considered negligible. The assumption is reasonable, based on the findings from Section 4.5, p. 41. The current 3D-model assumes ideal diamond meshes, therefore knot size was not accounted for. After the convergence study, the mesh with 2376 triangular elements and 3603 degrees of freedom was used. The comparison of section radiuses and distances from the reference section between the numerical and the experimental results is presented in Figure 5.8, p. 47 and Figure 5.9, p. 47. In the figures it is seen that the results of the numerical models correlate well with the experimental results. The biggest difference is observed comparing radiuses of the section at 24 meshes. It could be because the contact between the knots was not taken into account in neither of the models. For both models, the prediction for a control section distance is more accurate than

the prediction of the cod-end radius according to the Table 5.3, p. 48 and Table 5.4, p. 48. The distance predictions of the 3D-model are generally lower than the distance prediction of the axis-symmetric model due to absence of the knots in the model. None of the results is within the 95 % confidence interval, however, given the physical uncertainties it is deemed acceptable.



**Figure 5.8:** Raw data for the test 1: Section radiuses (left) and distances from the reference section (right).



**Figure 5.9:** Raw data for the test 2: Section radiuses (left) and distances from the reference section (right).

### 5.1.3 Conclusion

The data from 2 out of 4 experiments on the cod-end made of a single panel was presented. Considering the physical uncertainties, such as the unstable test structure, large marker size and calibration error the experiment shows that both numerical models were able to predict the correct equilibrium shape for the particular cod-end within an acceptable accuracy. The advantage of the axis-symmetric model for this particular case is that it is significantly less computationally demanding compared to the 3D-model, yet providing the results of the same quality. Inclusion of the drag force on the twines that is possible in 3D-model did not contribute to the improved accuracy due to its insignificance compared to the other forces.

**Table 5.3:** Quantitative results comparison for test 1. The experimentally obtained radiuses and distances are given with 95 % confidence interval. The percentile difference between the mean values of the experiment and numerical models is given.

<b>Section</b>	$R_{exp}$ [m]	$R_{axi}$ [m]	$R_{3D}$ [m]	$Err_{axi}$ [%]	$Err_{3D}$ [%]
24 meshes	$63.4 \pm 0.2$	42.0	45.3	33.7	28.5
36 meshes	$187.7 \pm 1.3$	168.0	178.5	10.5	4.9
43 meshes (catch)	$386.5 \pm 1.8$	402.0	404.5	-4.0	-4.7
<b>Section</b>	$L_{exp}$ [m]	$L_{axi}$ [m]	$L_{3D}$ [m]	$Err_{axi}$ [%]	$Err_{3D}$ [%]
24 meshes	$2394.8 \pm 0.6$	2405.0	2379.2	-0.4	0.7
36 meshes	$3611.9 \pm 1.1$	3655.0	3562.3	-1.2	1.4
43 meshes (catch)	$4229.1 \pm 1.6$	4301.0	4167.9	-1.7	1.4
End	$4674.0 \pm 2.3$	4751.0	4531.8	-1.6	3

**Table 5.4:** Quantitative results comparison for test 2. The experimentally obtained radiuses and distances are given with 95 % confidence interval. The percentile difference between the mean values of the experiment and numerical models is given.

<b>Section</b>	$R_{exp}$ [m]	$R_{axi}$ [m]	$R_{3D}$ [m]	$Err_{axi}$ [%]	$Err_{3D}$ [%]
24 meshes	$65.0 \pm 0.2$	43.0	45.2	33.8	30.4
36 meshes	$180.5 \pm 1.3$	168.0	178.3	6.9	1.2
43 meshes (catch)	$405.8 \pm 2.6$	402.0	404.2	0.9	0.3
<b>Section</b>	$L_{exp}$ [m]	$L_{axi}$ [m]	$L_{3D}$ [m]	$Err_{axi}$ [%]	$Err_{3D}$ [%]
24 meshes	$2439.1 \pm 0.6$	2405.0	2379.1	1.4	2.5
36 meshes	$3682.1 \pm 1.8$	3655.0	3562.3	0.7	3.3
43 meshes (catch)	$4332.2 \pm 1.4$	4302.0	4168.0	0.7	3.8
End	$4742.3 \pm 4.9$	4752.0	4532.2	-0.2	4.4

# Conclusion and recommendations for future work

---

In this project two numerical models of cod-end deformation were implemented and compared. Throughout both numerical and experimental comparison it was found that the axis-symmetric model with the hexagonal mesh representation can present a fast and reliable estimation for the deformation of small scale cod-ends. Small scale mostly implies the cod-ends with a small twine thickness, because for those the twine drag, opening resistance and bending resistance are negligible.

The experimental study explored only one case of the cod-end that was in the expected range of application for the axis-symmetric model. Validation of the models with thicker twines is required in order to support the results of the numerical comparison between the models, when the drag on the netting is considered. The 3D motion tracking system showed a great applicability in the quantitative experimental validation of the cod-ends. The application of the system can be realized to a bigger extent by performing experiments on cod-ends with selvages, whose shape is not axis-symmetric.

The implementation of both models followed the object-oriented programming method, which allows the extension of the code with the new features, for example, a triangular element based on the hexagonal mesh representation. In this project the guided user interface was not implemented, however, there is a possibility to do so if the code is desired to be developed into an actual in-house software for SINTEF OCEAN.

A big attention was given to the optimization of the solution method to find the equilibrium shape of the cod-end. It was found that the numerical axis-symmetric model could be combined with the 3D FEM model as a qualified initial guess to the more detailed calculation including additional mechanical behaviour. This cooperation was found to be favourable for the 3D FEM model, even when the axis-symmetric model was outside its range of validity. This idea can be further expanded by studying similar application of the other axis-symmetric cod-end models. The models such as the analytical axis-symmetric model by O'Neill or the numerical axis-symmetric model with diamond mesh representation by Priour could be used. The T90 model was implemented and tested, but not used as extensively as the T0 model. However, when the 3D model is upgraded with the triangular element with the hexagonal mesh representation, the application of the T90 model as an initial shape can be studied.

Apart from using the other numerical/analytical cod-end models as the quality initial shape there are alternative possibilities for the initial shape for those calculation cases, that can not be well approximated by the axis-symmetric models. Those are the cod-ends including selvages, straps or panels with more than one type of meshes. The approach suggest to start from a simple coarse initial shape and getting to a finer solution by switching between mesh refinement techniques and recalculation of the equilibrium shape. The method showed a good potential comparable to the initial shape based on the axis-symmetric model and could be optimized to yield a better performance. This optimization can take place by considering an adaptive structured mesh generation, that refines and smoothens the cod-end accounting for the curvature as well as the positions of round straps or selvages.

During the implementation process it was found that a slow math library did not allow to explore the 3D FEM model and its response do different inputs in a comfortable way. Therefore, it was critical to improve the calculation speed at least to some degree. The solution was found in an open source direct sparse matrix library CSPARSE.NET that allowed to step up the performance for both models. If desired, the calculation performance of the models can be pushed even further exploring the possibilities of native direct sparse, or native iterative sparse solvers. On the other hand, well working and robust code can continue being developed using a slower mathematics library, and then invest in a commercial package, that has a guarantee of high performance and robustness.

The robustness of the classic Newton-Raphson solution algorithm for both models was reinforced not only with the better initial shape, but also restart and line search techniques. Giving the positive results, there

---

is still the minor dependency on helper solver parameters, whose optimal value is not defined automatically. Other solution techniques shall be researched in the field of globally convergent optimization / non-linear system solution algorithms.

---

---

# Bibliography

---

- [1] Australian Fisheries Management Authority. *Trawling*. Read 1/10/2017. 2015. URL: <http://www.afma.gov.au/portfolio-item/trawling/>.
  - [2] Johann Heinrich von Thünen Institute. *Improving selectivity of trawls*. Read 14/05/2018. 2015. URL: <https://www.thuenen.de/en/of/projects/fisheries-and-survey-technology/improving-selectivity-of-trawls/>.
  - [3] N.A.H. Madsen N. Madsen K. Hansen. “Behavior of different trawl codend concepts”. In: (2015).
  - [4] J. H. B. Robertson P. A. M. Stewart. “Attachments to Codends”. In: (1980).
  - [5] F.G. O’Neil. “Selectivity in trawl fishing gears”. In: (2017).
  - [6] Michael Winther. *Optimization-based analysis of large cable-net structures for fishery*. Department of Mechanical Engineering, SINTEF Ocean: Aalborg University, 2009.
  - [7] D. Priour. “Modelling axis-symmetric codends made of hexagonal mesh types”. In: (2014).
  - [8] B. Herrmann D. Priour and F. G. O’Neill. “Modelling axisymmetric cod-ends made of different mesh types”. In: (2008).
  - [9] F.G. O’Neil B.Herrmann. “PRESEMO—a predictive model of codend selectivity—a tool for fishery managers”. In: (2007).
  - [10] O’Neil. “Differential equations governing the geometry of a diamond mesh cod-end of a trawl net”. In: (1997).
  - [11] Daniel Priour. *A finite element method for netting: application to fish cages and fishing gear*. Ifremer: Springer, 2013.
  - [12] D. Priour and F. G. O’Neill. “Comparison and validation of two models of netting deformation”. In: (2009).
  - [13] D. Fredriksson I. Tsukrov O. Eroshkin. “Finite element modeling of net panels using a consistent net element”. In: (2003).
  - [14] F. G. O’Neill. “Bending of twines and fibres under tension”. In: (2002).
  - [15] B Morvan D Priour Z Guede G Bles. “Finite element model for the assessment of the mesh resistance to opening of fishing nets”. In: (2016).
  - [16] Manuel González Amelia de la Prada. “Nonlinear stiffness models of a net twine to describe mesh resistance to opening of flexible net structures”. In: (2014).
  - [17] A. Sala et al. “Experimental method for quantifying resistance to the opening of netting panels”. In: (2007).
  - [18] Manuel González Amelia de la Prada. “Assessing the suitability of gradient-based energy minimization methods to calculate the equilibrium shape of netting structures”. In: (2014).
  - [19] Qualisys. *Qualisys motion capturing system*. Read 3/10/2017. 2015. URL: <https://www.qualisys.com/>.
  - [20] David Kincaid Ward Cheney. *Numerical Mathematics and Computing, Sixth edition*. Belmont, CA: Thomson Higher Education, 2004.
  - [21] M.J. Casarella Y.I. Choo. “Hydrodynamic resistance of towed cables”. In: (1971).
-

- [22] F. G. O'Neill and T. O'Donoghue. "The fluid dynamic loading on catch and the geometry of trawl cod-ends". In: (1997).
- [23] Scott W. Ambler. *UML 2 Class Diagrams: An Agile Introduction*. Read 28/05/2018. 2014. URL: <http://www.agilemodeling.com/artifacts/classDiagram.htm>.
- [24] ENVENIO. *Advantages of Structured Grids Over Unstructured Grids*. Read 29/05/2018. 2016. URL: <http://blog.ervenio.com/advantages-of-structured-grids-over-unstructured-grids>.
- [25] Jonathan Richard Shewchuk Christian Woltering. *A 2d Quality Mesh Generator and Delaunay Triangulator*. Read 29/05/2018. 2017. URL: <https://vsvv.org/contribution/triangle>.
- [26] Math.net project. *Math.NET Numerics Intel Math Kernel Library (MKL)*. Read 29/05/2018. 2017. URL: <https://numerics.mathdotnet.com/mkl.html>.
- [27] Tim Davis. *A concise library for solving sparse linear systems with direct methods*. Read 29/05/2018. 2006. URL: <https://github.com/wo80/CSparse.NET>.
- [28] Accord.NET Framework. *Machine learning, computer vision, statistics and general scientific computing for .NET*. Read 29/05/2018. 2017. URL: <https://github.com/accord-net/framework>.
- [29] C. T. Kelley. *Iterative Methods for Linear and Nonlinear Equations*. Philadelphia: Society for Industrial and Applied Mathematics, 1995.
- [30] Department of Aerospace Engineering Sciences University of Colorado at Boulder. "NFEM Chapter 35 Nonconservative Loading: Overview". In: (1999).
- [31] Robert Bridson. "Robustifying Newton, Constraints, Nonlinear Least Squares". In: (2008).
- [32] Jean E. Schweitzer. *Analysis and application of subdivision surfaces*. Seattle, Washington: University of Washington, 1996.
- [33] L. A. Krag B. Herrmann D. Priour. "Simulation-based study of the combined effect on cod-end size selection of turning meshes by 90 and reducing the number of meshes in the circumference for round fish". In: (2006).
- [34] Council of the European Union European Parliament. "Council Regulation (EC) No 2187/2005 of 21 December 2005 for the conservation of fishery resources through technical measures in the Baltic Sea, the Belts and the Sound, amending Regulation (EC) No 1434/98 and repealing Regulation (EC) No 88/98". In: (2005).
- [35] Ruye Wang. *Parabolic Interpolation*. Read 29/05/2018. 2015. URL: <http://fourier.eng.hmc.edu/e176/lectures/NM/node25.html>.
- [36] Matt's Webcorner Stanford University. *Subdivision*. Read 29/05/2018. 2014. URL: <https://graphics.stanford.edu/~mdfisher/subdivision.html>.
- [37] Wikipedia. *Polygon mesh*. Read 29/05/2018. 2018. URL: [https://en.wikipedia.org/wiki/Polygon\\_mesh](https://en.wikipedia.org/wiki/Polygon_mesh).
- [38] Luca Castelli Aleardi. *Mesh representations and data structures*. Read 29/05/2018. 2014. URL: <http://www.enseignement.polytechnique.fr/informatique/INF562/>.
- [39] The-Crankshaft Publishing's. *Mesh Processing (Advanced Methods in Computer Graphics) Part 4*. Read 29/05/2018. 2014. URL: <http://what-when-how.com/advanced-methods-in-computer-graphics/mesh-processing-advanced-methods-in-computer-graphics-part-4/>.
- [40] SAS IP Ansys. *Skewness*. Read 28/05/2018. 2014. URL: [https://www.sharcnet.ca/Software/Ansys/17.0/en-us/help/wb\\_msh/msh\\_skewness.html](https://www.sharcnet.ca/Software/Ansys/17.0/en-us/help/wb_msh/msh_skewness.html).

# Appendix





## APPENDIX A

---

# Derivatives for 3d FEM model

---

Definition of the forces and the tangent stiffness matrices in the numerical models relies heavily on the derivatives given below. Both Lagrange notation ( $'$ ) and Leibniz notation ( $\partial$ ) are used. The derivative of a scalar or a vector written in Lagrange notation means a derivative of its each component with respect to all the variables  $\mathbf{k}$ .

$$\mathbf{F}' = \frac{\partial \mathbf{F}}{\partial \mathbf{k}} \quad (\text{A.1})$$

where:

$$\mathbf{k} = (x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3)$$

$$d = (U_2 - U_1)(V_3 - V_1) - (U_3 - U_1)(V_2 - V_1) \quad (\text{A.2})$$

### Derivatives of twine vector components

Complete set of derivatives in  $\mathbf{U}'$  and  $\mathbf{V}'$  forms a 3 by 9 matrix.

$$\begin{aligned} \frac{\partial V_x}{\partial x_1} &= \frac{\partial V_y}{\partial y_1} = \frac{\partial V_z}{\partial z_1} = \frac{V_2 - V_3}{d} \\ \frac{\partial V_x}{\partial x_2} &= \frac{\partial V_y}{\partial y_2} = \frac{\partial V_z}{\partial z_2} = \frac{V_3 - V_1}{d} \\ \frac{\partial V_x}{\partial x_3} &= \frac{\partial V_y}{\partial y_3} = \frac{\partial V_z}{\partial z_3} = \frac{V_1 - V_2}{d} \end{aligned} \quad (\text{A.3})$$

For all other combinations of denominator and nominator, the derivative is 0.

$$\begin{aligned} \frac{\partial U_x}{\partial x_1} &= \frac{\partial U_y}{\partial y_1} = \frac{\partial U_z}{\partial z_1} = \frac{U_3 - U_2}{d} \\ \frac{\partial U_x}{\partial x_2} &= \frac{\partial U_y}{\partial y_2} = \frac{\partial U_z}{\partial z_2} = \frac{U_1 - U_3}{d} \\ \frac{\partial U_x}{\partial x_3} &= \frac{\partial U_y}{\partial y_3} = \frac{\partial U_z}{\partial z_3} = \frac{U_2 - U_1}{d} \end{aligned} \quad (\text{A.4})$$

For all other combinations of denominator and nominator, the derivative is 0.

### Derivatives of twine vector length

Complete set of derivatives in  $|\mathbf{U}'|$  and  $|\mathbf{V}'|$  forms a 1 by 9 matrix.

$$\frac{\partial |\mathbf{U}|}{\partial x_1} = \frac{U_x (V_2 - V_3)}{d |\mathbf{U}|} \quad \frac{\partial |\mathbf{V}|}{\partial x_1} = \frac{V_x (U_3 - U_2)}{d |\mathbf{V}|} \quad (\text{A.5})$$

$$\frac{\partial |\mathbf{U}|}{\partial y_1} = \frac{U_y (V_2 - V_3)}{d |\mathbf{U}|} \quad \frac{\partial |\mathbf{V}|}{\partial y_1} = \frac{V_y (U_3 - U_2)}{d |\mathbf{V}|}$$

$$\frac{\partial |\mathbf{U}|}{\partial z_1} = \frac{U_z (V_2 - V_3)}{d |\mathbf{U}|} \quad \frac{\partial |\mathbf{V}|}{\partial z_1} = \frac{V_z (U_3 - U_2)}{d |\mathbf{V}|} \quad (\text{A.6})$$


---

$$\begin{aligned}
\frac{\partial |\mathbf{U}|}{\partial x_2} &= \frac{U_x (V_3 - V_1)}{d|\mathbf{U}|} & \frac{\partial |\mathbf{V}|}{\partial x_2} &= \frac{V_x (U_1 - U_3)}{d|\mathbf{V}|} \\
\frac{\partial |\mathbf{U}|}{\partial y_2} &= \frac{U_y (V_3 - V_1)}{d|\mathbf{U}|} & \frac{\partial |\mathbf{V}|}{\partial y_2} &= \frac{V_y (U_1 - U_3)}{d|\mathbf{V}|} \\
\frac{\partial |\mathbf{U}|}{\partial z_2} &= \frac{U_z (V_3 - V_1)}{d|\mathbf{U}|} & \frac{\partial |\mathbf{V}|}{\partial z_2} &= \frac{V_z (U_1 - U_3)}{d|\mathbf{V}|} \\
\frac{\partial |\mathbf{U}|}{\partial x_3} &= \frac{U_x (V_1 - V_2)}{d|\mathbf{U}|} & \frac{\partial |\mathbf{V}|}{\partial x_3} &= \frac{V_x (U_2 - U_1)}{d|\mathbf{V}|} \\
\frac{\partial |\mathbf{U}|}{\partial y_3} &= \frac{U_y (V_1 - V_2)}{d|\mathbf{U}|} & \frac{\partial |\mathbf{V}|}{\partial y_3} &= \frac{V_y (U_2 - U_1)}{d|\mathbf{V}|} \\
\frac{\partial |\mathbf{U}|}{\partial z_3} &= \frac{U_z (V_1 - V_2)}{d|\mathbf{U}|} & \frac{\partial |\mathbf{V}|}{\partial z_3} &= \frac{V_z (U_2 - U_1)}{d|\mathbf{V}|}
\end{aligned} \tag{A.7}$$

### Derivatives of unit twine vector

Complete set of derivatives in  $\hat{\mathbf{U}}$ ' and  $\hat{\mathbf{V}}$ ' forms a 3 by 9 matrix.

$$\hat{\mathbf{U}} = \frac{\mathbf{U}}{|\mathbf{U}|} \tag{A.8}$$

$$\frac{\partial \hat{\mathbf{U}}}{\partial \mathbf{k}} = \left( \frac{\partial \mathbf{U}}{\partial \mathbf{k}} |\mathbf{U}| - \mathbf{U} \frac{\partial |\mathbf{U}|}{\partial \mathbf{k}} \right) \frac{1}{|\mathbf{U}|^2} \tag{A.9}$$

### Second derivatives of twine vector length

Complete set of derivatives in  $|\mathbf{U}|''$  and  $|\mathbf{V}|''$  forms a 9 by 9 matrix.

$$\frac{\partial^2 |\mathbf{U}|}{\partial \mathbf{k}^2} = \frac{\partial \hat{\mathbf{U}}^T}{\partial \mathbf{k}} \cdot \frac{\partial \mathbf{U}}{\partial \mathbf{k}} \tag{A.10}$$

### Derivatives of twine tension forces

Complete set of derivatives in each  $\mathbf{F}_1$ ,  $\mathbf{F}_2$ ,  $\mathbf{F}_3$  forms 3 by 9 matrix.

$$\frac{\partial \mathbf{F}_1}{\partial \mathbf{k}} = \frac{EA(V_3 - V_2)}{2} \left[ \frac{\partial \mathbf{U}}{\partial \mathbf{k}} \left( \frac{1}{l_0} - \frac{1}{|\mathbf{U}|} \right) \frac{\partial |\mathbf{U}|}{\partial \mathbf{k}} \frac{\mathbf{U}}{|\mathbf{U}|^2} \right] + \frac{EA(U_2 - U_3)}{2} \left[ \frac{\partial \mathbf{V}}{\partial \mathbf{k}} \left( \frac{1}{l_0} - \frac{1}{|\mathbf{V}|} \right) \frac{\partial |\mathbf{V}|}{\partial \mathbf{k}} \frac{\mathbf{V}}{|\mathbf{V}|^2} \right] \tag{A.11}$$

$$\frac{\partial \mathbf{F}_2}{\partial \mathbf{k}} = \frac{EA(V_1 - V_3)}{2} \left[ \frac{\partial \mathbf{U}}{\partial \mathbf{k}} \left( \frac{1}{l_0} - \frac{1}{|\mathbf{U}|} \right) \frac{\partial |\mathbf{U}|}{\partial \mathbf{k}} \frac{\mathbf{U}}{|\mathbf{U}|^2} \right] + \frac{EA(U_3 - U_1)}{2} \left[ \frac{\partial \mathbf{V}}{\partial \mathbf{k}} \left( \frac{1}{l_0} - \frac{1}{|\mathbf{V}|} \right) \frac{\partial |\mathbf{V}|}{\partial \mathbf{k}} \frac{\mathbf{V}}{|\mathbf{V}|^2} \right] \tag{A.12}$$

$$\frac{\partial \mathbf{F}_3}{\partial \mathbf{k}} = \frac{EA(V_2 - V_1)}{2} \left[ \frac{\partial \mathbf{U}}{\partial \mathbf{k}} \left( \frac{1}{l_0} - \frac{1}{|\mathbf{U}|} \right) \frac{\partial |\mathbf{U}|}{\partial \mathbf{k}} \frac{\mathbf{U}}{|\mathbf{U}|^2} \right] + \frac{EA(U_1 - U_2)}{2} \left[ \frac{\partial \mathbf{V}}{\partial \mathbf{k}} \left( \frac{1}{l_0} - \frac{1}{|\mathbf{V}|} \right) \frac{\partial |\mathbf{V}|}{\partial \mathbf{k}} \frac{\mathbf{V}}{|\mathbf{V}|^2} \right] \tag{A.13}$$

### Derivatives of drag force attack angles

Complete set of the drag force attack angle derivatives forms 1 by 9 matrix.

$$\alpha = \text{acos} \left( \frac{\mathbf{c} \cdot \mathbf{U}}{|\mathbf{c}| |\mathbf{U}|} \right) \quad \beta = \text{acos} \left( \frac{\mathbf{c} \cdot \mathbf{V}}{|\mathbf{c}| |\mathbf{V}|} \right) \tag{A.14}$$

$$\frac{\partial \alpha}{\partial \mathbf{k}} = -\frac{1}{\sin(\alpha)} \left( \frac{\mathbf{c}}{|\mathbf{c}|} \right)^T \frac{\partial \hat{\mathbf{U}}}{\partial \mathbf{k}} \quad \frac{\partial \beta}{\partial \mathbf{k}} = -\frac{1}{\sin(\beta)} \left( \frac{\mathbf{c}}{|\mathbf{c}|} \right)^T \frac{\partial \hat{\mathbf{V}}}{\partial \mathbf{k}} \tag{A.15}$$

### Derivatives of pressure and friction drag force magnitudes

Complete set of the drag force magnitude derivatives forms 1 by 9 matrix for both pressure and friction drag.

$$\frac{\partial |\mathbf{F}|}{\partial \mathbf{k}} = \rho_{water} C_d D m_0 |\mathbf{c}|^2 \cos(\alpha) \sin(\alpha) \frac{d}{2} \frac{\partial \alpha}{\partial \mathbf{k}} \quad (\text{A.16})$$

$$\frac{\partial |\mathbf{T}|}{\partial \mathbf{k}} = -C_f \rho_{water} C_d D m_0 |\mathbf{c}|^2 \cos(\alpha) \sin(\alpha) \frac{d}{2} \frac{\partial \alpha}{\partial \mathbf{k}} \quad (\text{A.17})$$

For the effects on  $\mathbf{V}$  twine angle  $\alpha$  is substituted  $\beta$ .

### Derivatives of the pressure drag direction

Complete set of the derivatives of the pressure drag direction  $\mathbf{E}$  forms 3 by 9 matrix for both pressure and friction drag.

$$\mathbf{E}_U = \mathbf{U} \times (\mathbf{c} \times \mathbf{U}) \quad (\text{A.18})$$

$$\frac{\partial \mathbf{E}_U}{\partial \mathbf{k}} = 2\mathbf{c} \cdot \left( \mathbf{U}^T \cdot \frac{\partial \mathbf{U}}{\partial \mathbf{k}} \right) - \mathbf{U} \cdot \left( \mathbf{c}^T \cdot \frac{\partial \mathbf{U}}{\partial \mathbf{k}} \right) - (\mathbf{U}^T \cdot \mathbf{c}) \cdot \frac{\partial \mathbf{U}}{\partial \mathbf{k}} \quad (\text{A.19})$$

Exactly the same is valid for  $\mathbf{E}_V$  component.

### Derivatives of the pressure drag

Complete set of derivatives in each  $\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3$  forms 3 by 9 matrix and is concatenated into 9 by 9 matrix.

$$\frac{\partial \mathbf{F}_1}{\partial \mathbf{k}} = \frac{\partial \mathbf{F}_2}{\partial \mathbf{k}} = \frac{\partial \mathbf{F}_3}{\partial \mathbf{k}} = \frac{1}{3} \left( \frac{\mathbf{E}}{|\mathbf{E}|} \frac{\partial |\mathbf{F}|}{\partial \mathbf{k}} + \frac{|\mathbf{F}|}{|\mathbf{E}|^2} \left( |\mathbf{E}| \frac{\partial \mathbf{E}}{\partial \mathbf{k}} - \frac{\mathbf{E}}{|\mathbf{E}|} \left( \mathbf{E} \frac{\partial \mathbf{E}_U}{\partial \mathbf{k}} \right) \right) \right) \quad (\text{A.20})$$

Exactly the same is valid for the contribution of  $V$  twines.

### Derivatives of the friction drag

Complete set of derivatives in each  $\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3$  forms 3 by 9 matrix and is concatenated into 9 by 9 matrix.

$$\begin{aligned} \frac{\partial \mathbf{T}_1}{\partial \mathbf{k}} = \frac{\partial \mathbf{T}_2}{\partial \mathbf{k}} = \frac{\partial \mathbf{T}_3}{\partial \mathbf{k}} = & \frac{1}{3} \frac{\mathbf{T}}{|\mathbf{T}|} \frac{\partial |\mathbf{T}|}{\partial \mathbf{k}} + \frac{1}{3} \frac{|\mathbf{T}|}{|\cos(\alpha)| |\mathbf{U}|} \left( \cos(\alpha) \frac{\partial \mathbf{U}}{\partial \mathbf{k}} + \sin(\alpha) \mathbf{U} \frac{\partial \alpha}{\partial \mathbf{k}} \right) - \\ & - \frac{1}{3} \frac{\mathbf{T}}{|\cos(\alpha)| |\mathbf{U}|} \left( \frac{|\cos(\alpha)|}{|\mathbf{U}|} \mathbf{U} \frac{\partial \mathbf{U}}{\partial \mathbf{k}} - \sin(\alpha) |\mathbf{U}| \frac{\cos(\alpha)}{|\cos(\alpha)|} \frac{\partial \mathbf{U}}{\partial \mathbf{k}} \right) \end{aligned} \quad (\text{A.21})$$

### Derivatives of mesh opening angle

Complete set of the derivatives of the mesh opening angle  $\alpha$  forms 1 by 9 matrix the first derivative and 9 by 9 matrix for the second derivative.

$$\alpha = \frac{1}{2} \arccos \left( \frac{\mathbf{U} \cdot \mathbf{V}}{|\mathbf{U}| |\mathbf{V}|} \right) \quad (\text{A.22})$$

$$\frac{\partial \alpha}{\partial \mathbf{k}} = -\frac{1}{2 \sin(2\alpha)} \left( \frac{\partial \hat{\mathbf{U}}^T}{\partial \mathbf{k}} \cdot \frac{\mathbf{V}}{|\mathbf{V}|} + \frac{\partial \hat{\mathbf{V}}^T}{\partial \mathbf{k}} \cdot \frac{\mathbf{U}}{|\mathbf{U}|} \right) \quad (\text{A.23})$$

$$\frac{\partial^2 \alpha}{\partial \mathbf{k}^2} = \frac{A'B - B'A}{B^2} \quad (\text{A.24})$$

where:

$$A = \mathbf{V}^T \cdot \frac{\partial \mathbf{U}}{\partial \mathbf{k}} - \frac{\mathbf{U}^T \cdot \mathbf{V}}{|\mathbf{U}|} \frac{\partial |\mathbf{U}|}{\partial \mathbf{k}} + \mathbf{U}^T \cdot \frac{\partial \mathbf{V}}{\partial \mathbf{k}} - \frac{\mathbf{U}^T \cdot \mathbf{V}}{|\mathbf{V}|} \frac{\partial |\mathbf{V}|}{\partial \mathbf{k}} B = -2 \sin(2\alpha) |\mathbf{U}| |\mathbf{V}|$$

$$\begin{aligned}
A' &= \frac{\partial |\mathbf{U}|^T}{\partial \mathbf{k}} \cdot \frac{\partial |\mathbf{V}|}{\partial \mathbf{k}} - \left( \frac{\partial \hat{\mathbf{U}}^T}{\partial \mathbf{k}} \cdot \mathbf{V} - \frac{\partial \mathbf{V}^T}{\partial \mathbf{k}} \cdot \mathbf{U} \right) \frac{1}{|\mathbf{U}|} \frac{\partial |\mathbf{U}|}{\partial \mathbf{k}} - \frac{\mathbf{U}^T \cdot \mathbf{V}}{|\mathbf{U}|} \frac{\partial^2 |\mathbf{U}|}{\partial \mathbf{k}^2} + \\
&+ \frac{\partial |\mathbf{V}|^T}{\partial \mathbf{k}} \cdot \frac{\partial |\mathbf{U}|}{\partial \mathbf{k}} - \left( \frac{\partial \hat{\mathbf{V}}^T}{\partial \mathbf{k}} \cdot \mathbf{U} - \frac{\partial \mathbf{U}^T}{\partial \mathbf{k}} \cdot \mathbf{V} \right) \frac{1}{|\mathbf{V}|} \frac{\partial |\mathbf{V}|}{\partial \mathbf{k}} - \frac{\mathbf{U}^T \cdot \mathbf{V}}{|\mathbf{U}|} \frac{\partial^2 |\mathbf{V}|}{\partial \mathbf{k}^2} \\
B' &= -4\cos(2\alpha) \frac{\partial \alpha}{\partial \mathbf{k}} |\mathbf{U}| |\mathbf{V}| - 2\sin(2\alpha) \left( \frac{\partial |\mathbf{U}|^T}{\partial \mathbf{k}} |\mathbf{V}| + \frac{\partial |\mathbf{V}|^T}{\partial \mathbf{k}} |\mathbf{U}| \right)
\end{aligned}$$

### Derivatives of the mesh opening resistance

Complete set of derivatives of the mesh opening resistance forms 9 by 9 matrix.  $H$  is mesh resistance to opening,  $d$  is the amount of knots in the triangle and  $\alpha_0$  is the initial mesh openness.

$$\frac{\partial \mathbf{F}}{\partial \mathbf{k}} = Hd \frac{\partial \alpha^T}{\partial \mathbf{k}} \frac{\partial \alpha}{\partial \mathbf{k}} + Hd(\alpha - \alpha_0) \frac{\partial^2 \alpha}{\partial \mathbf{k}^2} \quad (\text{A.25})$$

### Derivatives of bending angles

$$\alpha = \frac{1}{2} \arccos \left( \frac{\mathbf{U}_a \cdot \mathbf{U}_b}{|\mathbf{U}_a| |\mathbf{U}_b|} \right) \quad \beta = \frac{1}{2} \arccos \left( \frac{\mathbf{V}_a \cdot \mathbf{V}_b}{|\mathbf{V}_a| |\mathbf{V}_b|} \right)$$

The first and the second derivatives of the bending angle are found in the same way as the derivatives of the mesh opening angles by substituting the pair of  $\mathbf{U}$  and  $\mathbf{V}$  by  $\mathbf{U}_a$  and  $\mathbf{U}_b$  or  $\mathbf{V}_a$  and  $\mathbf{V}_b$  depending on the desired direction.

### Derivatives of the bending curvature

Complete set of derivatives of the curvature forms 1 by 9 matrix. Curvature is the inverse of the curvature radius  $R$ . The same applies for the  $V$  component.

$$\begin{aligned}
A'_U &= n_{Ua} \hat{\mathbf{U}}_a \frac{\partial \mathbf{U}_a}{\partial k} \\
B'_U &= n_{Ub} \hat{\mathbf{U}}_b \frac{\partial \mathbf{U}_b}{\partial k} \\
C'_U &= \frac{n_{Ua} \mathbf{U}_a + n_{Ub} \mathbf{U}_b}{|n_{Ua} \mathbf{U}_a + n_{Ub} \mathbf{U}_b|} \left( n_{Ua} \frac{\partial \mathbf{U}_a}{\partial k} + n_{Ub} \frac{\partial \mathbf{U}_b}{\partial k} \right) \\
ABC'_U &= A'_U B_U C_U + A_U B'_U C_U + A_U B_U C'_U \\
S'_U &= \frac{A'_U A_U B_U^2 + B'_U A_U^2 B_U - A'_U A_U^3 + B'_U B_U C_U^2 + C'_U B_U^2 C_U - B'_U B_U^3 + A'_U A_U C_U^2 + C'_U A_U^2 C_U - C'_U C_U^3}{8S_U} \\
\kappa_U &= 4 \frac{S'_U A_U * B_U C_U - ABC'_U S_U}{(A_U B_U C_U)^2}
\end{aligned} \quad (\text{A.26})$$

### Derivatives of bending resistance

Complete set of derivatives of the mesh opening resistance forms 12 by 12 matrix.  $EI$  is the bending stiffness and  $|V_4 - V_3|$  is the length of the common edge in  $UV$  coordinates and  $\kappa$  is the curvature or the inverse of the curvature radius  $R$ . The same applies for the  $V$  component.

$$\frac{\partial \mathbf{F}}{\partial \mathbf{k}} = -EI |V_4 - V_3| \left( \kappa^T \frac{\partial \alpha}{\partial \mathbf{k}} + \kappa \frac{\partial^2 \alpha}{\partial \mathbf{k}^2} \right) \quad (\text{A.27})$$

---

**Derivatives of catch pressure forces**

Complete set of derivatives in each  $\mathbf{F}_1$ ,  $\mathbf{F}_2$ ,  $\mathbf{F}_3$  forms 3 by 9 matrix and is concatenated into 9 by 9 matrix.  $\mathbf{c}$  is the water speed vector.

$$p = \frac{1}{2} \rho_{water} C_d |\mathbf{c}|^2 \quad (\text{A.28})$$

$$\frac{\partial \mathbf{F}_1}{\partial x_1} = \frac{p}{6} \begin{pmatrix} 0 \\ z_3 - z_2 \\ y_2 - y_3 \end{pmatrix} \quad \frac{\partial \mathbf{F}_1}{\partial y_1} = \frac{p}{6} \begin{pmatrix} z_2 - z_3 \\ 0 \\ x_3 - x_2 \end{pmatrix} \quad \frac{\partial \mathbf{F}_1}{\partial z_1} = \frac{p}{6} \begin{pmatrix} y_3 - y_2 \\ x_2 - x_3 \\ 0 \end{pmatrix} \quad (\text{A.29})$$

$$\frac{\partial \mathbf{F}_1}{\partial x_2} = \frac{p}{6} \begin{pmatrix} 0 \\ z_1 - z_3 \\ y_3 - y_1 \end{pmatrix} \quad \frac{\partial \mathbf{F}_1}{\partial y_2} = \frac{p}{6} \begin{pmatrix} z_3 - z_1 \\ 0 \\ x_1 - x_3 \end{pmatrix} \quad \frac{\partial \mathbf{F}_1}{\partial z_2} = \frac{p}{6} \begin{pmatrix} y_1 - y_3 \\ x_3 - x_1 \\ 0 \end{pmatrix} \quad (\text{A.30})$$

$$\frac{\partial \mathbf{F}_1}{\partial x_3} = \frac{p}{6} \begin{pmatrix} 0 \\ z_2 - z_1 \\ y_1 - y_2 \end{pmatrix} \quad \frac{\partial \mathbf{F}_1}{\partial y_3} = \frac{p}{6} \begin{pmatrix} z_1 - z_2 \\ 0 \\ x_2 - x_1 \end{pmatrix} \quad \frac{\partial \mathbf{F}_1}{\partial z_3} = \frac{p}{6} \begin{pmatrix} y_2 - y_1 \\ x_1 - x_2 \\ 0 \end{pmatrix} \quad (\text{A.31})$$

The derivatives of the nodal forces  $\mathbf{F}_2$  and  $\mathbf{F}_3$  are calculated with the same expressions.

---



## APPENDIX B

# Derivatives for the axis-symmetric models

### Twine tension Jacobian

Tensile force of a spring can be represented with the following equation:

$$\mathbf{F} = \frac{EA}{l_0} (|\mathbf{x}_{ij}| - l_0) \hat{\mathbf{x}}_{ij} \quad (\text{B.1})$$

Where  $\mathbf{x}_{ij} = \mathbf{x}_i - \mathbf{x}_j$  is the vector between 2 nodes,  $\hat{\mathbf{x}}_{ij}$  is a unit or normalized vector and  $l_0$  is the initial twine length. The Jacobian of these tension forces is calculated as follows:

$$\frac{\partial \mathbf{F}}{\partial \mathbf{x}_i} = \frac{EA}{l_0} \left[ (|\mathbf{x}_{ij}| - l_0) \frac{\partial \hat{\mathbf{x}}_{ij}}{\partial \mathbf{x}_i} + \hat{\mathbf{x}}_{ij} \frac{\partial |\mathbf{x}_{ij}|}{\partial \mathbf{x}_i} \right] \quad (\text{B.2})$$

The derivatives of the vector magnitude and the normalized vector are obtained as follows:

$$\frac{\partial |\mathbf{x}_{ij}|}{\partial \mathbf{x}_i} = \left( \frac{\mathbf{x}_{ij}}{|\mathbf{x}_{ij}|} \right)^T \frac{\partial \mathbf{x}_{ij}}{\partial \mathbf{x}_i} = \hat{\mathbf{x}}_{ij}^T \frac{\partial \mathbf{x}_{ij}}{\partial \mathbf{x}_i} \quad (\text{B.3})$$

$$\frac{\partial \hat{\mathbf{x}}_{ij}}{\partial \mathbf{x}_i} = \frac{1}{|\mathbf{x}_{ij}|^2} \left( |\mathbf{x}_{ij}| \frac{\partial \mathbf{x}_{ij}}{\partial \mathbf{x}_i} - \mathbf{x}_{ij} \frac{\partial |\mathbf{x}_{ij}|}{\partial \mathbf{x}_i} \right) = \frac{1}{|\mathbf{x}_{ij}|} \left( \frac{\partial \mathbf{x}_{ij}}{\partial \mathbf{x}_i} - \hat{\mathbf{x}}_{ij} \cdot \hat{\mathbf{x}}_{ij}^T \frac{\partial \mathbf{x}_{ij}}{\partial \mathbf{x}_i} \right) \quad (\text{B.4})$$

Substituting the derivatives of the vector norm and the derivatives of the unit vector yields the following expression:

$$\begin{aligned} \frac{\partial \mathbf{F}}{\partial \mathbf{x}_i} &= \frac{EA}{l_0} \left[ (|\mathbf{x}_{ij}| - l_0) \frac{1}{|\mathbf{x}_{ij}|} \left( \frac{\partial \mathbf{x}_{ij}}{\partial \mathbf{x}_i} - \hat{\mathbf{x}}_{ij} \cdot \hat{\mathbf{x}}_{ij}^T \frac{\partial \mathbf{x}_{ij}}{\partial \mathbf{x}_i} \right) + \hat{\mathbf{x}}_{ij} \cdot \hat{\mathbf{x}}_{ij}^T \frac{\partial \mathbf{x}_{ij}}{\partial \mathbf{x}_i} \right] = \\ &= \frac{EA}{l_0} \left[ \left( 1 - \frac{l_0}{|\mathbf{x}_{ij}|} \right) (\mathbf{I} - \hat{\mathbf{x}}_{ij} \cdot \hat{\mathbf{x}}_{ij}^T) + \hat{\mathbf{x}}_{ij} \cdot \hat{\mathbf{x}}_{ij}^T \right] \frac{\partial \mathbf{x}_{ij}}{\partial \mathbf{x}_i} \end{aligned} \quad (\text{B.5})$$

The derivatives of the vector  $\mathbf{x}_{ij}$  should be calculated accounting for whether it is the T0 model (cylindrical coordinates) or the T90 model (Cartesian coordinates).

### Catch pressure Jacobian

The Jacobian of the catch pressure forces is a block diagonal matrix where each block is 2 by 6 (for T0) and corresponds to each node in the meridian. It shows the change in the nodal forces in  $x$  and  $r$  directions due to the change in the coordinates of the node as well as its two neighbours. The block of the Jacobian is exemplified for the node  $b_i$  whose neighbours are node  $a_i$  and  $c_i$ .

$$\frac{\partial \mathbf{Fb}}{\partial \mathbf{x}_b} = \begin{bmatrix} 0 & -0.5\theta(a_{ir} + b_{ir})P \\ \theta a_{ir}P & -0.5\theta(a_{ix} - b_{ix})P \\ 0 & 0.5\theta(a_{ir} + b_{ir})P - 0.5\theta(b_{ir} + c_{ir})P \\ 0 & -0.5\theta(a_{ix} - b_{ix})P - 0.5\theta(b_{ix} - c_{ix})P \\ 0 & 0.5\theta(b_{ir} + c_{ir})P \\ -\theta c_{ir}P & -0.5\theta(b_{ix} - c_{ix})P \end{bmatrix}^T \quad (\text{B.6})$$

where:

$$\mathbf{x}_b = (a_x, a_r, b_x, b_r, c_x, c_r)$$

For the T90 model the catch pressure forces Jacobian comes in blocks of 3 by 9 and is calculated in the similar fashion. The only difference is the transformation of the coordinates from cylindrical to the Cartesian system.

---

## APPENDIX C

---

# Line search algorithm

---

The solution of the system of non-linear equation  $\mathbf{F}(\mathbf{X}_i) = \mathbf{0}$  is obtained by the Newton-Raphson iterative method:

$$\mathbf{h}_i = -\mathbf{J}^{-1}(\mathbf{X}_i) \mathbf{F}(\mathbf{X}_i) \quad (\text{C.1})$$

$$\mathbf{X}_{i+1} = \mathbf{X}_i + \lambda \mathbf{h}_i \quad (\text{C.2})$$

The length of the step  $\lambda$  along NR direction that leads to the decrease in force residual can be found applying a line search method[29]. For a search direction  $\mathbf{h}_i$  a set of successive step lengths  $\lambda$  is tested until the sufficient decrease condition Eq. (C.3) is satisfied. The the decrease by a factor  $\beta = 10^{-4}$  is typically accepted.

$$|\mathbf{F}(\mathbf{X}_i + \lambda \mathbf{h}_i)| < (1 - \beta) |\mathbf{F}(\mathbf{X}_i)| \quad (\text{C.3})$$

The Armijo rule it is suggested a testing strategy that divides the step length  $\lambda$  by 2 after each unsuccessful trial. The choice of the factor 2 is not justified and is just a rule of thumb. When there are more than 2 unsuccessful trials the data from previous attempts is available for application of a smarter method called *3-point safeguarded parabolic interpolation*. This method suggests a new  $\lambda$  that minimizes an interpolating polynomial based on the previous trial steps. The motivation for this is that some problems respond well to one or two reductions in the step length by modest amounts (such as 0.5) and others require many such reductions, but might respond well to a more aggressive step length reduction (by factors of 0.1, say) [29].

After 2 rejected iterations there are 3 data points available with  $\lambda$  and corresponding force residual. The data points correspond to the result of the previous NR iteration, full step result of this iteration as well as half step result of this iteration as suggested by Armijo Rule.

$$\text{Point 1: } \lambda_0 = 0.0 \quad \text{and} \quad f(0)$$

$$\text{Point 2: } \lambda_p = 1.0 \quad \text{and} \quad f(\lambda_p)$$

$$\text{Point 3: } \lambda_c = 0.5 \quad \text{and} \quad f(\lambda_c)$$

$$\text{where:} \quad f(\lambda) = |\mathbf{F}(\mathbf{X}_i + \lambda \mathbf{h}_i)|^2 \quad (\text{C.4})$$

The subscript  $p$  stands for previously considered  $\lambda$  and subscript  $c$  stands for currently considered  $\lambda$ . The function for a parabola  $p(\lambda)$  to interpolate  $f(\lambda)$  reads:

$$p(\lambda) = f(0) + \frac{\lambda}{\lambda_p - \lambda_c} \left( \frac{(\lambda - \lambda_c)(f(\lambda_p) - f(0))}{\lambda_p} + \frac{(\lambda_p - \lambda)(f(\lambda_c) - f(0))}{\lambda_c} \right) \quad (\text{C.5})$$

If  $p''(0) > 0$  then there is a minimum for the parabola that takes the value:

$$\lambda_{min} = -\frac{p'(0)}{p''(0)} \quad (\text{C.6})$$

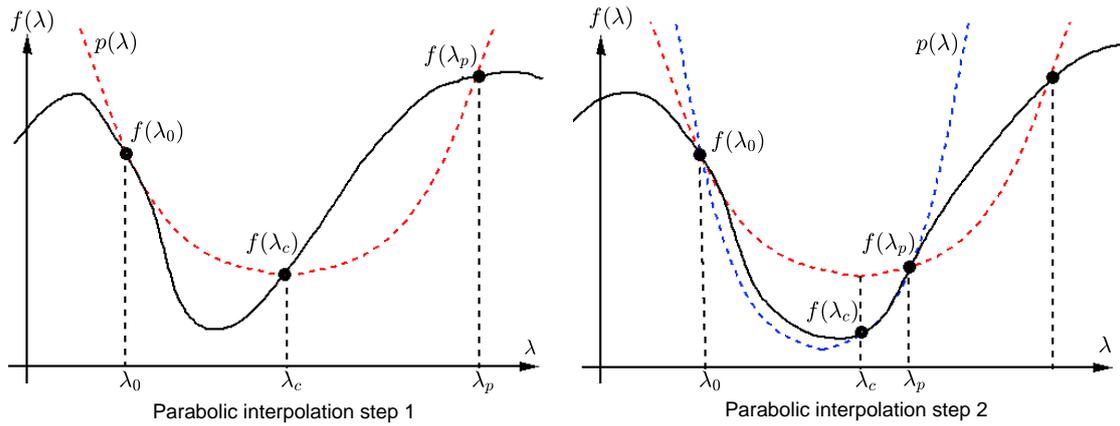
When the new minimum is found it is becomes a current point and the current point becomes the previous point as show in Figure C.1, p. xii. The search is repeated until the reduction condition in Eq. (C.3) is satisfied. There is a danger that the minimum maybe too near  $\lambda = 0$  to be of much use and, in fact, the iteration

---

may stagnate as a result. Therefore the parabolic interpolation method is extended with the *safeguarding* technique as shown in Eq. (C.8). Typically  $\sigma_0 = 0.1$  and  $\sigma_1 = 0.5$ .

$$\lambda_p = \lambda_c \tag{C.7}$$

$$\lambda_c = \begin{cases} \sigma_0 \lambda_c & \text{if } \lambda_{min} < \sigma_0 \lambda_c \\ \sigma_1 \lambda_c & \text{if } \lambda_{min} > \sigma_1 \lambda_c \text{ or } p''(0) < 0 \\ \lambda_{min} & \text{if } p''(0) > 0 \end{cases} \tag{C.8}$$



**Figure C.1:** Two iterations of the 3-point parabolic interpolation [35].

## APPENDIX D

---

# Numerical tests on the axis-symmetric model

---

In this appendix 3 editions of the axis-symmetric model were tested:

1. Simple cylindrical initial shape from [7] and classic Newton-Raphson algorithm. Further referred to as *V1*.
2. Simple cylindrical initial shape from [7] but the Newton-Raphson algorithm improved with the line search described in Appendix C. Further referred to as *V2*.
3. Improved cylindrical initial shape with smooth end from, the Newton-Raphson algorithm improved with the line search and restart techniques described in Section 3.4, p. 28. Further referred to as *V3*.

At the time of the test a managed dense solver from ACCORD.NET [28] math library was used which is much slower than the solver suggested in Section 3.4, p. 28. The purpose of the test was to identify the combinations of input parameters that make the solution algorithm slow down, diverge or converge to an incorrect tangled solution. Every new edition of the axis-symmetric model was supplied with the tools that are believed to make it more robust. In the tests involving the influence of added diagonal stiffness  $\alpha$  its value was kept low, in order to rely on its stabilizing effects as little as possible. The high  $\alpha$  value damps the quadratic convergence of the NR method, therefore in general it is desired to use either low  $\alpha$  or not use it at all. The results are presented in the following tables. The notations are explained in Table D.1, p. xiv.

1. Influence of  $\alpha$  and axial rigidity 1 on T0 Table D.2, p. xv and T90 Table D.11, p. xx.
2. Influence of  $\alpha$  and axial rigidity 2 on T0 Table D.3, p. xv and T90 Table D.12, p. xx.
3. Influence of  $\alpha$  and axial rigidity 3 on T0 Table D.4, p. xvi and T90 Table D.13, p. xxi.
4. Influence of the amount of catch on T0 Table D.5, p. xvi and T90 Table D.14, p. xxi.
5. Influence of the water speed on T0 Table D.6, p. xvii and T90 Table D.15, p. xxii.
6. Influence of the twine lengths and knot lengths on T0 Table D.7, p. xviii and T90 Table D.16, p. xxiii.
7. Influence of the twine lengths and knot lengths with different  $\alpha$  on T0 Table D.8, p. xviii and T90 Table D.17, p. xxiii.
8. Influence of the difference between the twine and knot lengths on T0 Table D.9, p. xix and T90 Table D.19, p. xxiv.
9. Influence of the difference between the twine and knot lengths with different  $\alpha$  on T0 Table D.9, p. xix and T90 Table D.19, p. xxiv.

Following ways of avoiding the convergence failure were undertaken. Divergence can be solved by setting an upper limit for the force residual. Once the limit is exceeded the algorithm is restarted with higher added stiffness. Incorrect solution can be detected by checking whether there are negative coordinates in the resulting meridian shape. Also the maximum radius and the maximum length of the final shape cannot go beyond the limit of fully stretched fully closed meshes. A better solution was not found for cycling behaviour other than defining a maximum number of iterations, after which the algorithm will restart with higher additional stiffness.

---

**Table D.1:** Notations.

Notation	Description
Type	Cod-end model type with either T0 or T90 mesh orientation
$n_x$	Number of meshes along the cod-end
$n_r$	Number of meshes in the cod-end circumference
$n_c$	Number of mesh rows blocked by catch
$r_0$	Entrance radius in [m]
$m_0$	Twine length of the mesh in [m]
$l_0$	Knot length of the mesh in [m]
$k_m$	Twine axial rigidity $EA$ in [N]
$k_l$	Knot axial rigidity $EA$ in [N]
$c$	Water speed in [ $\text{m s}^{-1}$ ]
$\alpha$	Additional diagonal stiffness as described in Section 3.4, p. 28 and [31] [ $\text{N m}^{-1}$ ]
Iter	Amount of iterations for convergence
Time	Elapsed time in [min:sec] format
DIV	Failure of the method due to divergence
INC	Failure of the method due to convergence to incorrect solution Figure D.1, p. xiv
CYC	Failure of the method due to cycling around between the same set of values unable to continue

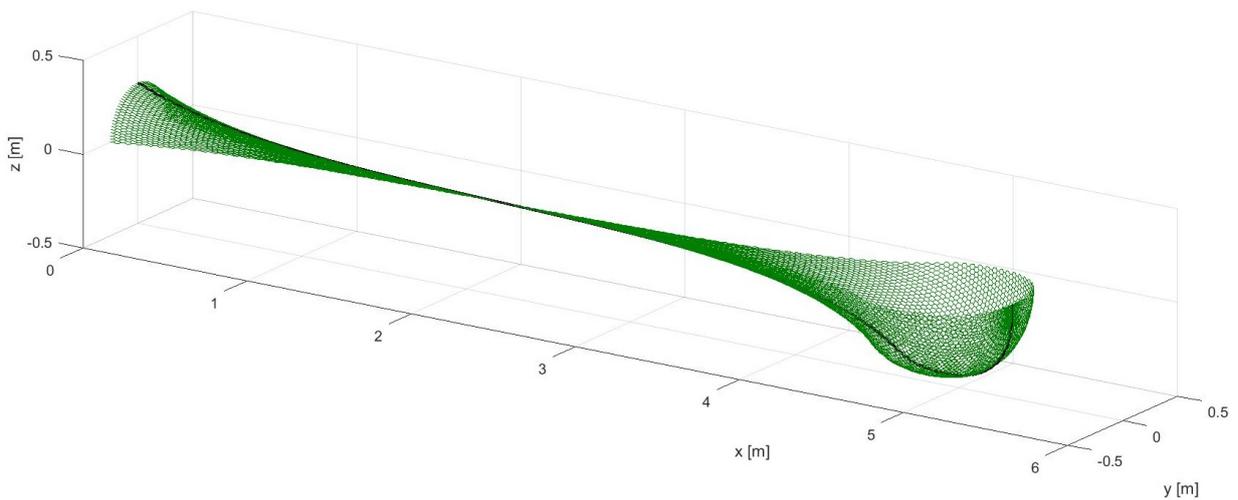
**Figure D.1:** Incorrect result of the iterative solution that satisfies the force equilibrium from Table D.8, p. xviii.

Table D.2: Influence of  $\alpha$  and axial rigidity part 1.

T0 Model parameters											V1		V2		V3	
Test	$n_x$	$n_r$	$n_c$	$r_0$	$l_0$	$m_0$	$k_l$	$k_m$	$c$	$\alpha$	Iter	Time	Iter	Time	Iter	Time
1.01	100	100	20	0.25	0.015	0.015	450	450	1.5	0.01	INC		17	00:04	10	00:02
1.02	100	100	20	0.25	0.015	0.015	450	450	1.5	0.1	13	00:03	18	00:04	13	00:03
1.03	100	100	20	0.25	0.015	0.015	450	450	1.5	1	48	00:12	33	00:07	26	00:06
1.04	100	100	20	0.25	0.015	0.015	450	450	1.5	2	85	00:21	52	00:12	39	00:09
1.05	100	100	20	0.25	0.015	0.015	450	450	1.5	5	197	00:49	84	00:19	74	00:20
1.06	100	100	20	0.25	0.015	0.015	450	450	1.5	10	383	01:37	109	00:25	113	00:30
1.07	100	100	20	0.25	0.015	0.015	450	450	1.5	15	570	02:25	134	00:31	124	00:32

Table D.3: Influence of  $\alpha$  and axial rigidity part 2.

T0 Model parameters											V1		V2		V3	
Test	$n_x$	$n_r$	$n_c$	$r_0$	$l_0$	$m_0$	$k_l$	$k_m$	$c$	$\alpha$	Iter	Time	Iter	Time	Iter	Time
2.01	100	100	20	0.25	0.015	0.015	4500	4500	1.5	0.01	DIV		INC		32	00:07
2.02	100	100	20	0.25	0.015	0.015	4500	4500	1.5	0.1	DIV		50	00:11	24	00:05
2.03	100	100	20	0.25	0.015	0.015	4500	4500	1.5	1	53	00:13	51	00:12	58	00:13
2.04	100	100	20	0.25	0.015	0.015	4500	4500	1.5	2	94	00:23	129	00:30	69	00:16
2.05	100	100	20	0.25	0.015	0.015	4500	4500	1.5	5	212	00:53	108	00:25	70	00:16
2.06	100	100	20	0.25	0.015	0.015	4500	4500	1.5	10	400	01:42	152	00:35	278	01:04
2.07	100	100	20	0.25	0.015	0.015	4500	4500	1.5	15	584	02:31	201	00:46	318	01:37

Table D.4: Influence of  $\alpha$  and axial rigidity part 3.

Test	$n_x$	$n_r$	$n_c$	T0 Model parameters						V1		V2		V3		
				$r_0$	$l_0$	$m_0$	$k_l$	$k_m$	$c$	$\alpha$	Iter	Time	Iter	Time	Iter	Time
3.01	100	100	20	0.25	0.015	0.015	45000	45000	1.5	0.01	DIV	134	00:31	38	00:10	
3.02	100	100	20	0.25	0.015	0.015	45000	45000	1.5	0.1	DIV	65	00:15	59	00:15	
3.03	100	100	20	0.25	0.015	0.015	45000	45000	1.5	1	DIV	81	00:18	79	00:20	
3.04	100	100	20	0.25	0.015	0.015	45000	45000	1.5	2	DIV	163	00:38	148	00:38	
3.05	100	100	20	0.25	0.015	0.015	45000	45000	1.5	5	230	00:58	183	00:42	137	00:36
3.06	100	100	20	0.25	0.015	0.015	45000	45000	1.5	10	442	01:51	292	01:07	195	00:51
3.07	100	100	20	0.25	0.015	0.015	45000	45000	1.5	15	651	02:44	772	02:59	218	00:58

Table D.5: Influence of the amount of catch.

Test	$n_x$	$n_r$	$n_c$	T0 Model parameters						V1		V2		V3		
				$r_0$	$l_0$	$m_0$	$k_l$	$k_m$	$c$	$\alpha$	Iter	Time	Iter	Time	Iter	Time
4.01	100	100	10	0.25	0.015	0.015	450	450	1.5	1	68	00:17	49	00:11	41	00:11
4.02	100	100	20	0.25	0.015	0.015	450	450	1.5	1	48	00:12	33	00:07	26	00:07
4.03	100	100	30	0.25	0.015	0.015	450	450	1.5	1	42	00:10	31	00:07	19	00:05
4.04	100	100	40	0.25	0.015	0.015	450	450	1.5	1	55	00:13	38	00:08	25	00:06
4.05	100	100	50	0.25	0.015	0.015	450	450	1.5	1	69	00:17	44	00:10	33	00:08
4.06	100	100	60	0.25	0.015	0.015	450	450	1.5	1	85	00:21	50	00:11	38	00:10
4.07	100	100	70	0.25	0.015	0.015	450	450	1.5	1	102	00:25	54	00:12	49	00:13
4.08	100	100	80	0.25	0.015	0.015	450	450	1.5	1	120	00:30	58	00:13	59	00:15
4.09	100	100	90	0.25	0.015	0.015	450	450	1.5	1	139	00:35	58	00:14	67	00:17
4.10	100	100	100	0.25	0.015	0.015	450	450	1.5	1	171	00:43	63	00:14	77	00:20

Table D.6: Influence of the water speed.

Test	T0 Model parameters										V1		V2		V3	
	$n_x$	$n_r$	$n_c$	$r_0$	$l_0$	$m_0$	$k_l$	$k_m$	$c$	$\alpha$	Iter	Time	Iter	Time	Iter	Time
5.01	100	100	20	0.25	0.015	0.015	450	450	0.5	1	287	01:12	266	01:06	43	00:11
5.02	100	100	20	0.25	0.015	0.015	450	450	0.6	1	210	00:54	111	00:26	40	00:10
5.03	100	100	20	0.25	0.015	0.015	450	450	0.7	1	162	00:41	62	00:14	40	00:10
5.04	100	100	20	0.25	0.015	0.015	450	450	0.8	1	130	00:33	125	00:29	39	00:10
5.05	100	100	20	0.25	0.015	0.015	450	450	0.9	1	170	00:28	95	00:22	42	00:11
5.06	100	100	20	0.25	0.015	0.015	450	450	1	1	90	00:23	74	00:17	36	00:09
5.07	100	100	20	0.25	0.015	0.015	450	450	1.1	1	77	00:20	58	00:13	35	00:09
5.08	100	100	20	0.25	0.015	0.015	450	450	1.2	1	76	00:17	47	00:11	34	00:09
5.09	100	100	20	0.25	0.015	0.015	450	450	1.3	1	59	00:15	42	00:09	33	00:08
5.10	100	100	20	0.25	0.015	0.015	450	450	1.4	1	53	00:13	37	00:08	32	00:08
5.11	100	100	20	0.25	0.015	0.015	450	450	1.5	1	48	00:12	33	00:07	26	00:06
5.12	100	100	20	0.25	0.015	0.015	450	450	1.6	1	43	00:11	30	00:07	22	00:05
5.13	100	100	20	0.25	0.015	0.015	450	450	1.7	1	40	00:10	25	00:06	20	00:05
5.14	100	100	20	0.25	0.015	0.015	450	450	1.8	1	37	00:09	24	00:06	20	00:05
5.15	100	100	20	0.25	0.015	0.015	450	450	1.9	1	34	00:08	23	00:05	19	00:06
5.16	100	100	20	0.25	0.015	0.015	450	450	2	1	32	00:08	22	00:05	18	00:04
5.17	100	100	20	0.25	0.015	0.015	450	450	2.1	1	30	00:07	20	00:04	17	00:04
5.18	100	100	20	0.25	0.015	0.015	450	450	2.2	1	28	00:07	19	00:04	16	00:04
5.19	100	100	20	0.25	0.015	0.015	450	450	2.3	1	DIV		19	00:04	16	00:04
5.20	100	100	20	0.25	0.015	0.015	450	450	2.4	1	DIV		18	00:04	16	00:04
5.21	100	100	20	0.25	0.015	0.015	450	450	2.5	1	DIV		19	00:04	16	00:05
5.22	100	100	20	0.25	0.015	0.015	450	450	2.6	1	DIV		21	00:04	16	00:04
5.23	100	100	20	0.25	0.015	0.015	450	450	2.7	1	DIV		16	00:04	15	00:04
5.24	100	100	20	0.25	0.015	0.015	450	450	2.8	1	DIV		17	00:04	14	00:03
5.25	100	100	20	0.25	0.015	0.015	450	450	2.9	1	DIV		18	00:04	14	00:03
5.26	100	100	20	0.25	0.015	0.015	450	450	3	1	DIV		18	00:04	14	00:04

Table D.7: Influence of the twine lengths and knot lengths.

Test	$n_x$	$n_r$	$n_c$	T0 Model parameters						V1		V2		V3		
				$r_0$	$l_0$	$m_0$	$k_l$	$k_m$	$c$	$\alpha$	Iter	Time	Iter	Time	Iter	Time
6.01	100	100	20	0.25	0.1	0.1	4500	4500	1.5	10	INC	67	00:16	43	00:11	
6.02	100	100	20	0.25	0.075	0.075	4500	4500	1.5	10	112	00:29	43	00:10	33	00:10
6.03	100	100	20	0.25	0.05	0.05	4500	4500	1.5	10	130	00:33	50	00:11	45	00:13
6.04	100	100	20	0.25	0.025	0.025	4500	4500	1.5	10	248	01:03	151	00:35	150	00:49
6.05	100	100	20	0.25	0.01	0.01	4500	4500	1.5	10	590	02:40	225	00:53	511	02:22

Table D.8: Influence of the twine lengths and knot lengths with different  $\alpha$ .

Test	$n_x$	$n_r$	$n_c$	T0 Model parameters						V1		V2		V3		
				$r_0$	$l_0$	$m_0$	$k_l$	$k_m$	$c$	$\alpha$	Iter	Time	Iter	Time	Iter	Time
7.01	100	100	20	0.25	0.1	0.1	4500	4500	1.5	1	DIY	156	00:37	95	00:22	
7.02	100	100	20	0.25	0.075	0.075	4500	4500	1.5	1	DIY	INC	104	00:24		
7.03	100	100	20	0.25	0.05	0.05	4500	4500	1.5	1	DIY	65	00:15	80	00:18	
7.04	100	100	20	0.25	0.025	0.025	4500	4500	1.5	1	DIY	45	00:10	42	00:09	
7.05	100	100	20	0.25	0.01	0.01	4500	4500	1.5	1	77	00:20	118	00:27	105	00:24

**Table D.9:** Influence of the difference between the twine and knot lengths.

Test	T0 Model parameters										V1		V2		V3	
	$n_x$	$n_r$	$n_c$	$r_0$	$l_0$	$m_0$	$k_l$	$k_m$	$c$	$\alpha$	Iter	Time	Iter	Time	Iter	Time
8.01	100	100	20	0.25	0.015	0.015	450	450	1.5	10	383	01:39	109	00:26	113	00:27
8.02	100	100	20	0.25	0.01125	0.015	450	450	1.5	10	402	01:49	108	00:25	126	00:33
8.03	100	100	20	0.25	0.0075	0.015	450	450	1.5	10	427	01:52	106	00:25	130	00:33
8.04	100	100	20	0.25	0.00375	0.015	450	450	1.5	10	458	02:02	124	00:29	129	00:32
8.05	100	100	20	0.25	0.0015	0.015	450	450	1.5	10	DIV		107	00:25	165	00:37
8.06	100	100	20	0.25	0.00075	0.015	450	450	1.5	10	488	02:11	108	00:25	172	00:44
8.07	100	100	20	0.25	0.00015	0.015	450	450	1.5	10	512	02:12	109	00:25	179	00:47

**Table D.10:** Influence of the difference between the twine and knot lengths with different  $\alpha$ .

Test	T0 Model parameters										V1		V2		V3	
	$n_x$	$n_r$	$n_c$	$r_0$	$l_0$	$m_0$	$k_l$	$k_m$	$c$	$\alpha$	Iter	Time	Iter	Time	Iter	Time
9.01	100	100	20	0.25	0.015	0.015	450	450	1.5	1	48	00:12	33	00:07	26	00:06
9.02	100	100	20	0.25	0.01125	0.015	450	450	1.5	1	49	00:12	31	00:07	38	00:08
9.03	100	100	20	0.25	0.0075	0.015	450	450	1.5	1	DIV		58	00:14	29	00:06
9.04	100	100	20	0.25	0.00375	0.015	450	450	1.5	1	DIV		64	00:15	50	00:11
9.05	100	100	20	0.25	0.0015	0.015	450	450	1.5	1	INC		242	00:57	37	00:08
9.06	100	100	20	0.25	0.00075	0.015	450	450	1.5	1	DIV		241	00:57	62	00:14
9.07	100	100	20	0.25	0.00015	0.015	450	450	1.5	1	DIV		70	00:16	51	00:11

Table D.11: Influence of  $\alpha$  and axial rigidity part 1.

Test	$n_x$	$n_r$	$n_c$	$r_0$	T90 Model parameters						V1		V2		V3	
					$l_0$	$m_0$	$k_l$	$k_m$	$c$	$\alpha$	Iter	Time	Iter	Time	Iter	Time
1.01	100	100	20	0.25	0.015	0.015	450	450	1.5	0.01	INC	00:03	80	00:13	34	00:04
1.02	100	100	20	0.25	0.015	0.015	450	450	1.5	0.1	18	00:02	22	00:03	36	00:04
1.03	100	100	20	0.25	0.015	0.015	450	450	1.5	1	98	00:14	38	00:05	26	00:03
1.04	100	100	20	0.25	0.015	0.015	450	450	1.5	2	185	00:25	35	00:05	32	00:03
1.05	100	100	20	0.25	0.015	0.015	450	450	1.5	5	449	01:04	58	00:10	32	00:03
1.06	100	100	20	0.25	0.015	0.015	450	450	1.5	10	889	02:21	94	00:15	41	00:05
1.07	100	100	20	0.25	0.015	0.015	450	450	1.5	15	1328	03:27	132	00:19	43	00:05

Table D.12: Influence of  $\alpha$  and axial rigidity part 2.

Test	$n_x$	$n_r$	$n_c$	$r_0$	T90 Model parameters						V1		V2		V3	
					$l_0$	$m_0$	$k_l$	$k_m$	$c$	$\alpha$	Iter	Time	Iter	Time	Iter	Time
2.01	100	100	20	0.25	0.015	0.015	4500	4500	1.5	0.01	DIV		INC		68	00:08
2.02	100	100	20	0.25	0.015	0.015	4500	4500	1.5	0.1	DIV		81	00:10	33	00:03
2.03	100	100	20	0.25	0.015	0.015	4500	4500	1.5	1	DIV		127	00:17	39	00:04
2.04	100	100	20	0.25	0.015	0.015	4500	4500	1.5	2	221	00:37	274	00:40	55	00:07
2.05	100	100	20	0.25	0.015	0.015	4500	4500	1.5	5	477	01:09	241	00:33	87	00:11
2.06	100	100	20	0.25	0.015	0.015	4500	4500	1.5	10	906	02:21	212	00:28	154	00:17
2.07	100	100	20	0.25	0.015	0.015	4500	4500	1.5	15	1368	03:13	349	00:52	175	00:22

**Table D.13:** Influence of  $\alpha$  and axial rigidity part 3.

Test	T90 Model parameters										V1		V2		V3	
	$n_x$	$n_r$	$n_c$	$r_0$	$l_0$	$m_0$	$k_l$	$k_m$	$c$	$\alpha$	Iter	Time	Iter	Time	Iter	Time
3.01	100	100	20	0.25	0.015	0.015	45000	45000	1.5	0.01	DIV	CYC	115	00:13		
3.02	100	100	20	0.25	0.015	0.015	45000	45000	1.5	0.1	DIV	141	00:18	102	00:11	
3.03	100	100	20	0.25	0.015	0.015	45000	45000	1.5	1	DIV	543	01:18	114	00:14	
3.04	100	100	20	0.25	0.015	0.015	45000	45000	1.5	2	323	00:50	360	00:50	85	00:10
3.05	100	100	20	0.25	0.015	0.015	45000	45000	1.5	5	495	00:54	623	01:30	251	00:29
3.06	100	100	20	0.25	0.015	0.015	45000	45000	1.5	10	974	01:54	1469	03:35	314	00:36
3.07	100	100	20	0.25	0.015	0.015	45000	45000	1.5	15	1440	03:35	1484	03:29	475	00:54

**Table D.14:** Influence of the amount of catch.

Test	T90 Model parameters										V1		V2		V3	
	$n_x$	$n_r$	$n_c$	$r_0$	$l_0$	$m_0$	$k_l$	$k_m$	$c$	$\alpha$	Iter	Time	Iter	Time	Iter	Time
4.01	100	100	10	0.25	0.015	0.015	450	450	1.5	1	430	00:57	58	00:07	41	00:04
4.02	100	100	20	0.25	0.015	0.015	450	450	1.5	1	98	00:15	38	00:04	26	00:02
4.03	100	100	30	0.25	0.015	0.015	450	450	1.5	1	44	00:05	101	00:14	68	00:07
4.04	100	100	40	0.25	0.015	0.015	450	450	1.5	1	32	00:04	80	00:11	31	00:03
4.05	100	100	50	0.25	0.015	0.015	450	450	1.5	1	33	00:04	48	00:06	43	00:04
4.06	100	100	60	0.25	0.015	0.015	450	450	1.5	1	37	00:04	29	00:03	33	00:04
4.07	100	100	70	0.25	0.015	0.015	450	450	1.5	1	43	00:05	27	00:03	30	00:04
4.08	100	100	80	0.25	0.015	0.015	450	450	1.5	1	53	00:08	29	00:04	37	00:04
4.09	100	100	90	0.25	0.015	0.015	450	450	1.5	1	DIV		31	00:05	40	00:04
4.10	100	100	100	0.25	0.015	0.015	450	450	1.5	1	DIV		32	00:05	41	00:04

Table D.15: Influence of the water speed.

Test	T90 Model parameters										V1		V2		V3	
	$n_x$	$n_r$	$n_c$	$r_0$	$l_0$	$m_0$	$k_l$	$k_m$	$c$	$\alpha$	Iter	Time	Iter	Time	Iter	Time
5.01	100	100	20	0.25	0.015	0.015	450	450	0.5	1	660	01:28	137	00:18	34	00:03
5.02	100	100	20	0.25	0.015	0.015	450	450	0.6	1	473	01:07	72	00:11	34	00:03
5.03	100	100	20	0.25	0.015	0.015	450	450	0.7	1	367	00:53	76	00:13	34	00:03
5.04	100	100	20	0.25	0.015	0.015	450	450	0.8	1	295	00:45	87	00:15	34	00:04
5.05	100	100	20	0.25	0.015	0.015	450	450	0.9	1	240	00:33	77	00:11	33	00:05
5.06	100	100	20	0.25	0.015	0.015	450	450	1	1	193	00:28	70	00:10	32	00:04
5.07	100	100	20	0.25	0.015	0.015	450	450	1.1	1	168	00:24	52	00:07	32	00:03
5.08	100	100	20	0.25	0.015	0.015	450	450	1.2	1	143	00:23	48	00:08	32	00:04
5.09	100	100	20	0.25	0.015	0.015	450	450	1.3	1	124	00:17	47	00:07	32	00:03
5.10	100	100	20	0.25	0.015	0.015	450	450	1.4	1	110	00:16	45	00:06	32	00:03
5.11	100	100	20	0.25	0.015	0.015	450	450	1.5	1	98	00:13	38	00:06	26	00:02
5.12	100	100	20	0.25	0.015	0.015	450	450	1.6	1	88	00:11	35	00:05	22	00:03
5.13	100	100	20	0.25	0.015	0.015	450	450	1.7	1	79	00:11	33	00:04	30	00:04
5.14	100	100	20	0.25	0.015	0.015	450	450	1.8	1	72	00:10	28	00:04	26	00:03
5.15	100	100	20	0.25	0.015	0.015	450	450	1.9	1	66	00:08	26	00:03	30	00:03
5.16	100	100	20	0.25	0.015	0.015	450	450	2	1	60	00:07	22	00:03	30	00:03
5.17	100	100	20	0.25	0.015	0.015	450	450	2.1	1	56	00:09	28	00:04	40	00:04
5.18	100	100	20	0.25	0.015	0.015	450	450	2.2	1	52	00:07	25	00:04	660	01:22
5.19	100	100	20	0.25	0.015	0.015	450	450	2.3	1	48	00:06	23	00:04	177	00:22
5.20	100	100	20	0.25	0.015	0.015	450	450	2.4	1	45	00:06	20	00:03	CYC	
5.21	100	100	20	0.25	0.015	0.015	450	450	2.5	1	42	00:05	20	00:02	39	00:04
5.22	100	100	20	0.25	0.015	0.015	450	450	2.6	1	39	00:05	18	00:02	106	00:11
5.23	100	100	20	0.25	0.015	0.015	450	450	2.7	1	37	00:06	16	00:02	43	00:06
5.24	100	100	20	0.25	0.015	0.015	450	450	2.8	1	35	00:06	16	00:02	124	00:15
5.25	100	100	20	0.25	0.015	0.015	450	450	2.9	1	33	00:04	14	00:02	41	00:04
5.26	100	100	20	0.25	0.015	0.015	450	450	3	1	31	00:04	18	00:02	184	00:22

Table D.16: Influence of the twine lengths and knot lengths.

Test	T90 Model parameters										V1		V2		V3	
	$n_x$	$n_r$	$n_c$	$r_0$	$l_0$	$m_0$	$k_l$	$k_m$	$c$	$\alpha$	Iter	Time	Iter	Time	Iter	Time
6.01	100	100	20	0.25	0.1	0.1	4500	4500	1.5	10	211	00:27	95	00:13	82	00:10
6.02	100	100	20	0.25	0.075	0.075	4500	4500	1.5	10	272	00:36	122	00:20	90	00:12
6.03	100	100	20	0.25	0.05	0.05	4500	4500	1.5	10	380	00:52	167	00:26	89	00:10
6.04	100	100	20	0.25	0.025	0.025	4500	4500	1.5	10	645	01:25	229	00:44	116	00:14
6.05	100	100	20	0.25	0.01	0.01	4500	4500	1.5	10	1200	02:38	682	01:41	50	00:06

Table D.17: Influence of the twine lengths and knot lengths with different  $\alpha$ .

Test	T90 Model parameters										V1		V2		V3	
	$n_x$	$n_r$	$n_c$	$r_0$	$l_0$	$m_0$	$k_l$	$k_m$	$c$	$\alpha$	Iter	Time	Iter	Time	Iter	Time
7.01	100	100	20	0.25	0.1	0.1	4500	4500	1.5	1	DIV		95		126	00:14
7.02	100	100	20	0.25	0.075	0.075	4500	4500	1.5	1	DIV		INC		144	00:16
7.03	100	100	20	0.25	0.05	0.05	4500	4500	1.5	1	DIV		167	00:39	CYC	
7.04	100	100	20	0.25	0.025	0.025	4500	4500	1.5	1	DIV	00:09	129	00:20	56	00:06
7.05	100	100	20	0.25	0.01	0.01	4500	4500	1.5	1	278	00:38	213	00:27	23	00:02

Table D.18: Influence of the difference between the twine and knot lengths.

Test	$n_x$	$n_r$	$n_c$	$r_0$	T90 Model parameters					V1		V2		V3		
					$l_0$	$m_0$	$k_l$	$k_m$	$c$	$\alpha$	Iter	Time	Iter	Time	Iter	Time
8.01	100	100	20	0.25	0.015	0.015	450	450	1.5	10	889	01:57	94	00:13	41	00:05
8.02	100	100	20	0.25	0.01125	0.015	450	450	1.5	10	752	01:56	630	01:50	133	00:18
8.03	100	100	20	0.25	0.0075	0.015	450	450	1.5	10	454	01:12	1018	03:18	41	00:05
8.04	100	100	20	0.25	0.00375	0.015	450	450	1.5	10	730	01:56	1929	05:53	62	00:07
8.05	100	100	20	0.25	0.0015	0.015	450	450	1.5	10	845	02:14	2321	07:10	102	00:11
8.06	100	100	20	0.25	0.00075	0.015	450	450	1.5	10	963	02:35	1828	05:03	113	00:14
8.07	100	100	20	0.25	0.00015	0.015	450	450	1.5	10	1063	02:37	1921	05:18	147	00:16

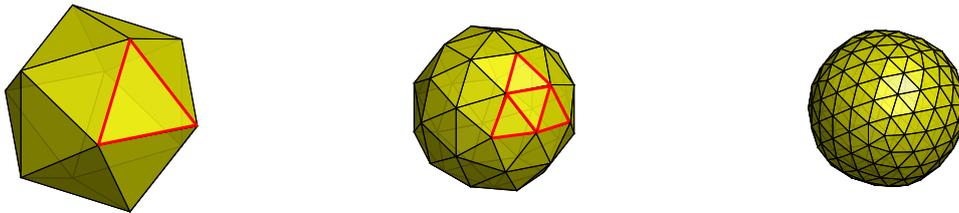
Table D.19: Influence of the difference between the twine and knot lengths with different  $\alpha$ .

Test	$n_x$	$n_r$	$n_c$	$r_0$	T90 Model parameters					V1		V2		V3		
					$l_0$	$m_0$	$k_l$	$k_m$	$c$	$\alpha$	Iter	Time	Iter	Time	Iter	Time
9.01	100	100	20	0.25	0.015	0.015	450	450	1.5	1	98	00:22	38	00:05	26	00:02
9.02	100	100	20	0.25	0.01125	0.015	450	450	1.5	1	DIV		193	00:38	102	00:14
9.03	100	100	20	0.25	0.0075	0.015	450	450	1.5	1	DIV		366	01:16	29	00:04
9.04	100	100	20	0.25	0.00375	0.015	450	450	1.5	1	DIV		588	02:09	27	00:03
9.05	100	100	20	0.25	0.0015	0.015	450	450	1.5	1	DIV		686	02:07	35	00:03
9.06	100	100	20	0.25	0.00075	0.015	450	450	1.5	1	DIV		1828	06:50	42	00:04
9.07	100	100	20	0.25	0.00015	0.015	450	450	1.5	1	DIV		1921	06:36	71	00:08

## APPENDIX E

# Mesh refinement and smoothing

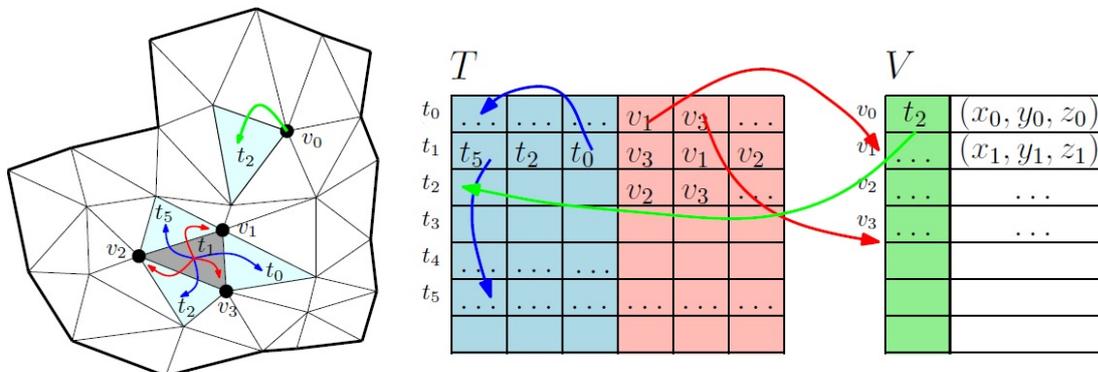
Subdivision is a powerful algorithm used, in its simplest application, to refine meshes. The underlying concepts are derived from spline refinement algorithms, but the idea is that there exists a well-defined smooth surface associated with any given input mesh (the exact surface depends on the subdivision algorithm used.) A refinement operation is the operation that takes the input mesh (control mesh) and generates an output mesh that is closer to the surface. If this refinement process is applied infinitely, the target surface will be achieved exactly. However, after just a few levels of refinement, the result is close enough to the limit surface that they are visually indistinguishable. Depending on the type of input mesh (triangular, quadrilateral, etc.) a different subdivision algorithm is used. Quadrilateral based meshes generally use Catmull-Clark, while triangular based meshes generally use Loop subdivision [36]. Subdivision is performed by manipulation on a mesh data structure.



**Figure E.1:** Refined shape of an icosahedron approaches the spherical limit surface.

### Mesh data structure

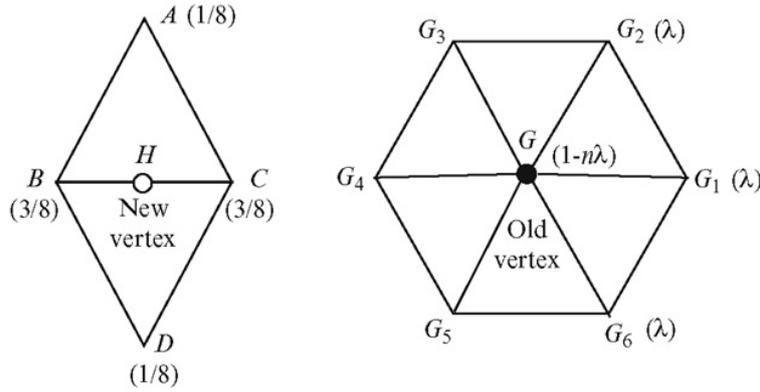
For the current application it was deemed sufficient to apply the simple face-based mesh structure due to its simple implementation. Besides there are other more advanced mesh data structures such as half-edge or winged-edge data structures [37]. For each triangle the basic version of the data structure stores: 3 references to neighboring faces, 3 references to incident vertices. For each vertex it stores 1 reference to an incident face as shown in Figure E.2, p. xxv. For further manipulation with edges, the data structure is expanded. For each edge the data structure stores 2 references to incident vertices and 2 references to the triangles sharing the edge. The triangle additionally stores 3 references to its 3 edges.



**Figure E.2:** Face-based mesh data structure [38].

### Loop subdivision

All subdivision algorithms start by replacing the geometric element (in our case, a triangle) with smaller versions of the same element. For every edge in the source mesh, a vertex is added and for every triangle on the mesh, the four triangles are created as shown in Figure E.1, p. xxv. The exact geometric location of these new edge vertices, and the new coordinates of the initial vertices, are all determined by the subdivision scheme. In the simple case, the coordinates can be left unchanged to correspond to the edge centers. In Loop scheme they are linear combinations of the neighbouring source mesh vertices [36]. Those linear combinations (subdivision masks) for the insertion of new vertices and transformation of existing vertices are shown in Figure E.3, p. xxvi.



**Figure E.3:** The weights for determining the spatial position of the newly created vertices as well as re-positioning of the old vertices in the mesh interior [39].

Insertion of new vertices:

$$p_H^{j+1} = \frac{1}{8}p_A^j + \frac{3}{8}p_B^j + \frac{3}{8}p_C^j + \frac{1}{8}p_D^j \quad (\text{E.1})$$

Transformation of the existing vertices requires the traversal operation to visit all the  $n$  vertices adjacent to the considered vertex only once. That is where the mesh data structure is essential.

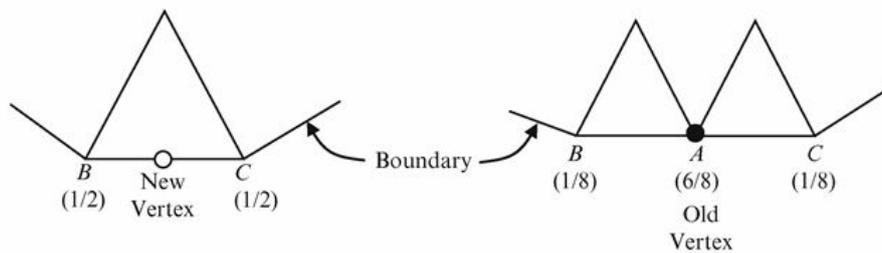
$$p_G^{j+1} = (1 - n\lambda)p_G^j + \lambda \sum_{i=1}^n p_{G_i}^j \quad (\text{E.2})$$

$$\lambda = \begin{cases} \frac{3}{16} & \text{if } n = 3 \\ \frac{3}{8n} & \text{otherwise} \end{cases} \quad (\text{E.3})$$

The boundaries are treated in the similar manner applying the subdivision mask shown in Figure E.4, p. xxvii

### Mesh smoothing by edge flipping

Considering two adjacent faces of a triangle mesh, there exist exactly two different configurations of the inner edge. Usually one of the configurations is more favourable in terms of triangle quality than the other. Triangle quality is defined by skewness, that shows how close to ideal shape (equilateral or equiangular) a face or cell is [40]. The idea of the edge flipping method is to evaluate the mean skewness in two possible edge configurations and accept the one with the least mean skewness. This manipulation can be performed for the entire mesh to yield the lower mesh skewness on the global level as shown in Figure E.5, p. xxvii.



**Figure E.4:** The weights for determining the spatial position of the newly created vertices as well as re-positioning of the old vertices on the mesh edge [39].

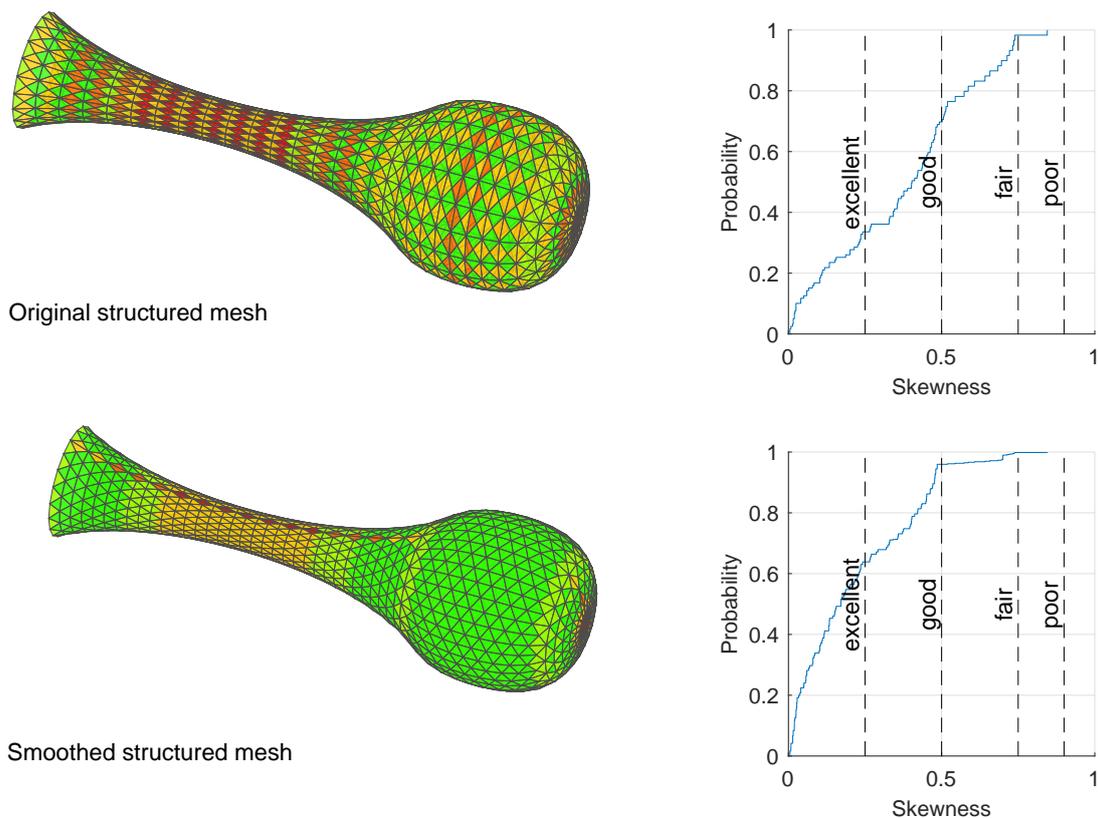
$$Skewness = \max\left(\frac{\theta_{max} - \theta_e}{180 - \theta_e}, \frac{\theta_e - \theta_{min}}{\theta_e}\right) \quad (E.4)$$

where:

$\theta_{max}$  largest angle in the face or cell

$\theta_{min}$  smallest angle in the face or cell

$\theta_e$  angle for an equiangular face/cell (e.g., 60 for a triangle, 90 for a square) (E.5)



**Figure E.5:** The effect of edge flipping on the FE mesh quality of the cod-end. The range of skewnesses according to [40].

In the figure it is seen that in the original structured mesh there is approximately 70 % of the finite elements that are *good* or *excellent*. After edge flipping there are more than 90 % of the *good* or *excellent*

elements with the majority in the *excellent* zone. Considered cod-end is made of two panels, and it is clearly seen that the edges on the panel joint are not flipped. This is due to the fact that flipping those edges will break the consistency in  $UV$  coordinates, since the considered pair of triangles belongs to two different panels.

---

APPENDIX F

---

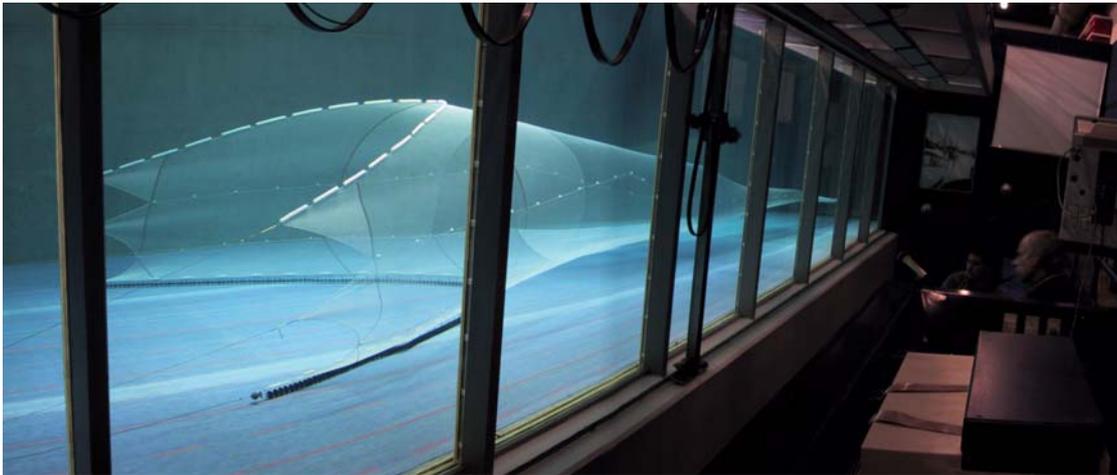
**SINTEF OCEAN flume tank in Hirtshals**

---

# The North Sea Centre Flume Tank

Managed and operated by SINTEF Fisheries and Aquaculture

Modern testing facility for fishing gear with a long history of servicing the net manufacturers and fishermen from all over the World



## Technical information:

### *Dimensions*

Over-all: L: 30, H: 6, W: 8 m  
 Measuring section 21.3, 2.7, 8 m.  
 Volume of water: 1200 m<sup>3</sup>.  
 Windows: 20 2x3 m

### *Propulsion*

Four propellers and motors of 64 KW generate the flow.

### *Velocity*

Maximum water speed: 1 m/s  
 Max. simulated towing speed in scale 1:5: 4,5 knots  
 Max. simulated towing speed in scale 1:20: 12 knots

### *Artificial bottom*

Conveyer belt type made of nylon. Speed adjustable, can be locked with water speed.

### *Equipment for measuring*

Object measuring: A full measurement of geometry in three dimensions is made by remote controlled video system.

Resistance and drag: Load cells using strain gauge technology. Series of load cells can measure strain from a few grams to 100 kg. Measurements are monitored by a data-logger system and converted to full-scale values.

### *Lights*

Above the tank: 18 x 400 W daylight floodlights.  
 Movable on a 3x2 m frame: 6 x 1.500 W floodlights.

### *Video*

Movable cameras are permanently displaying: Front view, in colour, top view (two cameras for spread measuring), side view (for height measuring). Professional studio camera for documenting tests. Video hardcopy printer making prints on paper from all the installed cameras.

### *Video conference*

The tank is fully equipped with a video mixer and fixed and movable cameras to transmit training, testing and demonstrations via videoconference to customers sitting in classrooms or videoconference studios abroad.

## Associated facilities

### *Net loft*

SINTEF has its own staff of skilled net makers and a net loft for constructing and altering the models.

### *Video editing*

Video film for instruction or advertising can be edited on DV-Cam video editing system with text generator. Films are converted into the type and system required by the customer.

**Activities and Services**

*Testing fishing gear*

Trawl manufacturers and research institutions from all over the world have tested more than 700 different designs of fishing gear; mainly trawls, but also Danish seines, Scottish seines and gill nets.

*Training and courses*

The flume tank is ideal for demonstrating the performance of fishing gear for an audience. SINTEF run workshops for fishermen, researchers, administrators and others interested to know about fisheries technology, fishing gear performance, selectivity of cod-end and trawls, etc.

*Research projects*

The flume tank is a vital instrument in a vast number of research projects.

*Testing other equipment*

The constant and even flow makes it possible to test other equipment than fishing gear, i.e. specific components of fishing gear, underwater vehicles, propellers etc.

**Costs 2004**

The hire is calculated on an hourly basis. Please ask for a quotation. A normal full day is around 3400 ECU (2004) including a staff of 3 (1 scientist, 2 technicians).

