Improvement of Rubble Ice Generation in Numerical Simulation of Ice Ridge and Structure Interaction

Adylio Vitarelli Neto

Master Thesis

presented in partial fulfillment
of the requirements for the double degree:
“Advanced Master in Naval Architecture” conferred by University of Liege
"Master of Sciences in Applied Mechanics, specialization in Hydrodynamics, Energetics and Propulsion” conferred by École Centrale de Nantes

developed at University of Rostock
in the framework of the

“EMSHIP”
Erasmus Mundus Master Course
in “Integrated Advanced Ship Design”

Ref. 159652-1-2009-1-BE-ERA MUNDUS-EMMC

Supervisor: Prof. Robert Bronsart, University of Rostock

Reviewer: Prof. Maciej Taczała, West Pomeranian University of Technology Szczecin

Rostock, February 2017

This page is intentionally left blank.

ABSTRACT

Vessels and offshore structures operating in the Arctic Ocean are expected to face ice ridges during operation. Ice ridges are formed when sea ice layers collide and break, creating a pile of rubble ice along a line. Experimental model tests in ice tanks are performed to assess loads on offshore structures interacting with ice ridges and the ability of a ship to break through an ice ridge.

Computational simulation of the ice ridge and structure interaction at the early design stage would allow the prediction of the loads and behaviour of the structure before testing in model ice basin. The numerical simulation under development at the Hamburg Ship Model Basin (HSVA) is capable of creating ice ridge and simulating the interaction with ships and offshore structures.

In the pre-existing version of the numerical simulation, the rubble ice particles were represented by three-dimensional rectangles. In order to improve the rubble ice geometry, and consequently, the interaction between ice-ice and ice-structure, this master thesis offers improvements on the geometry modeling, where they are represented by three-dimensional polygons.

The dimensions of the polygons in the simulation were obtained from samples of model scale rubble ice generated from ice ridges constructed at HSVA ice basin. The axis-aligned bounding box dimension and the number of edges of the rubble ice were collected. The data was fit in beta probabilistic distribution function (PDF). Thereafter, a PDF random variate of the axis-aligned bounding box and number of edges are the input for the creation of polygons in the simulation. In addition, the algorithm for the calculation of the volume, moment of inertia, superficial area and mass of the polygon shaped rubble ice was implemented and validated.

Subsequently, punch test simulation and ship simulation were carried out to compare the effects of the implementation with the pre-existing code. The software was validated with experimental model test data from HSVA.

This page is intentionally left blank.

CONTENTS

List of Figures

List of Tables

This page is intentionally left blank.

Declaration of Authorship

I declare that this thesis and the work presented in it are my own and have been generated by me as the result of my own original research.

Where I have consulted the published work of others, this is always clearly attributed.

Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.

I have acknowledged all main sources of help.

Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

This thesis contains no material that has been submitted previously, in whole or in part, for the award of any other academic degree or diploma.

I cede copyright of the thesis in favour of the University of Rostock.

Date: Signature

1. INTRODUCTION

The naval activity at the Arctic region consists of routes for transportation of goods through the Northern Sea Route and oil & gas exploitation area. Those activities require safe and economical operation due to the challenging conditions caused by the harsh environmental conditions.

The low temperature in the Arctic freezes the ocean layer forming sea ice. The sea ice is non-stationary, and because of this motion, ice ridges are created. New ridges are identified as first-year, those that survive one summer are named as second-year and those that survive more summers are named as multi-year ice ridges. The first-year ridges are one of the most important features to take into account when designing ice-going vessels to assess its capability to overcome ice ridges. The load on offshore structures interacting with ridges has to be evaluated for design purpose. The ice ridges are identified by meteorological services who publish the information in ice charts for ship route management.

To understand the vessel's behaviour on the ice environment, experimental models test in ice tanks and computational simulations are performed. This thesis presents improvements on rubble ice generation in the software under development at the Hamburg Ship Model Basin (HSVA), which is capable of simulating the interaction between vessels and offshore structures against ice ridges.

Previously, Seidel (2016) implemented the discrete element methods (DEM) the ice ridge creation and the punch test simulation. Alekseev (2016) introduced ship simulation into the code and discussed the influence of force coefficients in the results. Little information is known about the geometry of the rubble ice in ice tanks and their effects on the structures.

The thesis aim to introduce improvements of the rubble ice geometry in the code and analyse their effects on the results. To obtain the geometrical properties of the rubble ice generate in model scale ice ridges, measurements were carried out from ice ridges built at HSVA ice tank. The work includes the analysis of punch tests and ship simulation to validate the new features of the software.

1.1 Arctic Environment

According to Martin (2007), the Arctic Ocean has 79% contribution from the Atlantic Ocean through numerous straits while 19% comes from the Pacific Ocean through the Bering Strait. The origin of the remaining 2% are rivers.

The Arctic climate is highly influenced by the time of the year. During the summer, the surface air temperature has the order of magnitude of 10 °C. The peak of the winter occurs on

December 21st, so called polar night with temperatures of approximately -30 °C. The extreme low temperatures in which the Arctic Ocean is exposed to freezes a layer of ocean, forming sea ice.

The sea ice moves due to wind, ocean currents stresses on the ice sheet and interactions with another ice sheet. The results of sea ice motion are deformations of the ice layer by rafting and ridging.

Rafting occurs when the sea ice layers slide on top of the other. Ridging is the accumulation of floes that creates a pile of ice blocks in a line along the fracture of the ice layer. The thickness that the ice floe can reach in nature is between 5 and 25 cm (Martin, 2007). The thickness of the ice blocks is similar to the thickness of the surrounding level ice. A real ice ridge can be seen in Figure 1.

Figure 1. Ice ridge sail. Salvesen (1990)

The ice ridge structure is divided into three main components: the sail (above the waterline), the keel (below the waterline) and the consolidated layer (between sail and keel). The sail contains ice blocks and snow. The keel is a rubble pile of ice also known as an unconsolidated layer.

After the formation of a first-year ridge, the water that is trapped in the upper part of the ridge between the ice blocks freezes, forming the consolidated layer. The formation of an ice ridge and its nomenclature are illustrated in Figure 2. The main geometric parameters for the description of an ice ridge are as follow:

L_s – Length of sail

L_A – Length of consumed ice area

W_s – Width of sail

W_k – Width of keel

H_s – Height of sail

H_l – Height of level ice

H_k – Height of keel

H_f – Height of freeboard

H_d – Height of draft

α_k – Angle of keel

α_s – Angle of sail

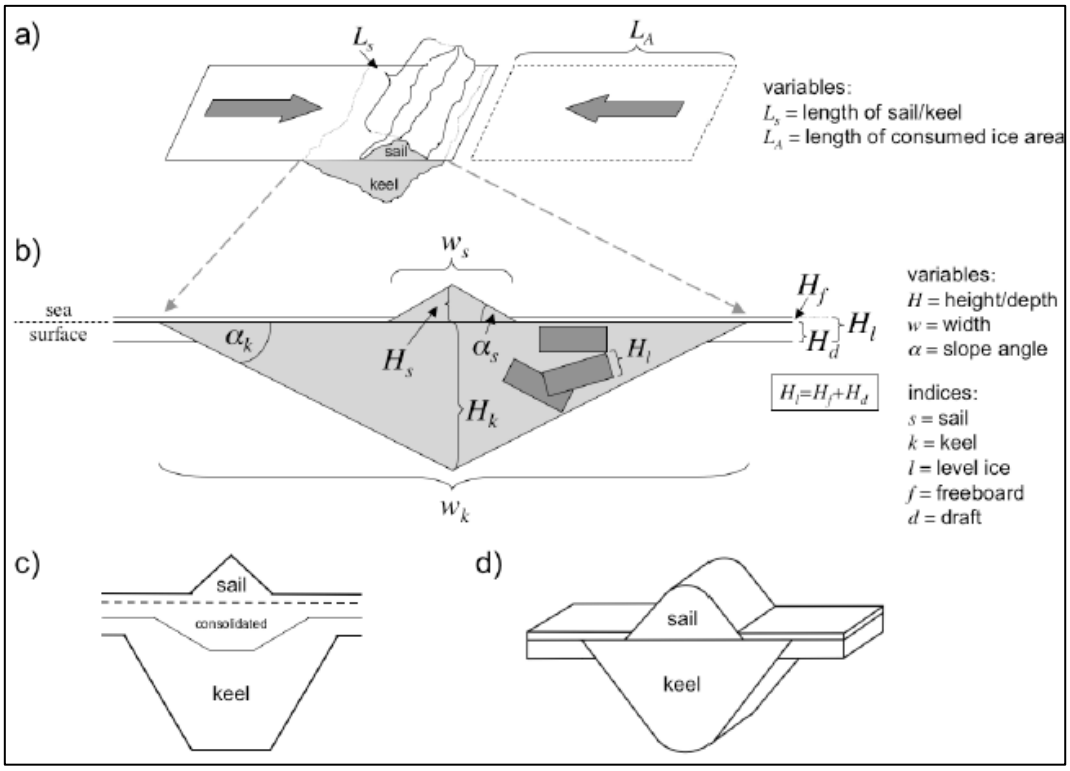

Figure 2. Ice ridge formation. Adapted from Martin (2007)

Another parameter to define the ice ridge is the keel porosity η, which is the relation between the volume of the ice ridge $V_{ridge\ keel}$ and the volume to the rubble ice V_{rubble_ice}. The pressure between ice rubble due to the buoyancy is proportional to the keel porosity.

$$\eta = \frac{V_{rubble\ ice}}{V_{ridge\ keel}} \tag{1}$$

1.2 Model Scale Test

Ice basins are able to recreate scaled ice ridges in a well-controlled environment to guarantee scaling properties according to Froude number. The Hamburg Ship Model Basin (HSVA) is capable to perform ship and offshore structure model test in their ice tank.

The ice ridge creation at the facility differs from the natural formation. First, a layer of level ice is generated with the desired thickness of the rubble ice. Then the ice is cut into stripes according to the final profile and porosity of the ridge. After cutting, the level ice is pushed by the main carriage against a steel beam while the stripes are pushed under the ice layer. Figure 3 shows a sketch of this process and the main dimensions of HSVA facility.

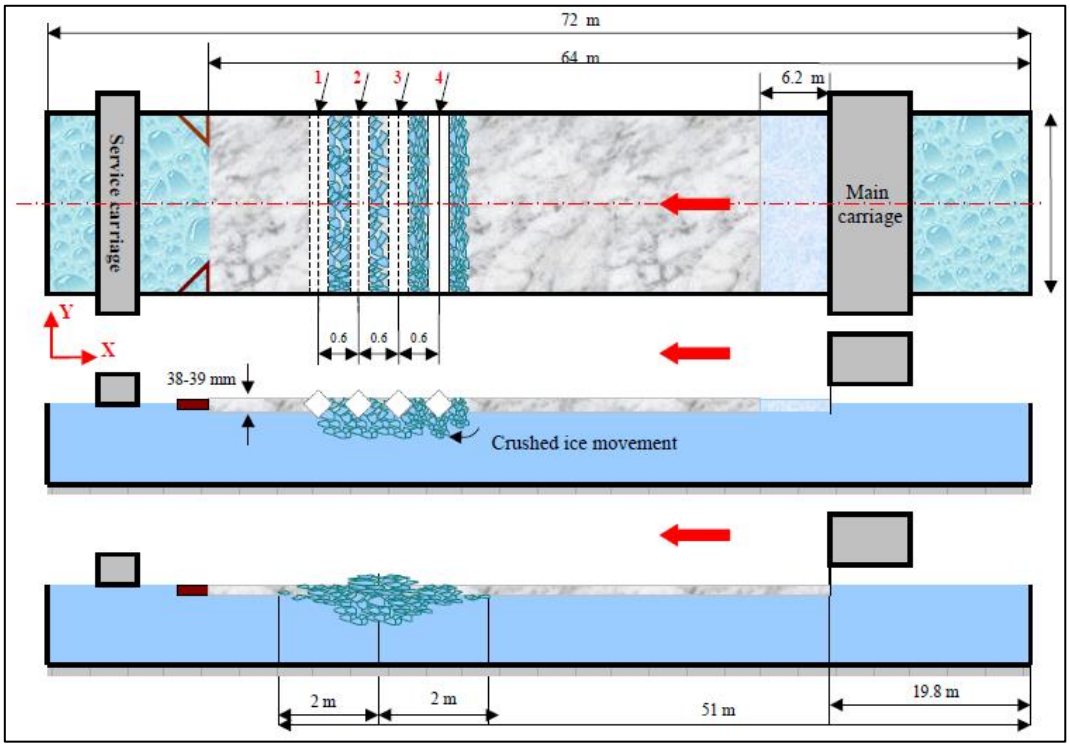

Figure 3. Ice ridge formation in model basin. Adapted from HSVA (2014).

ISO (2010) propose two options of testing technique of ship for ice model tanks, the first method is to either pull or push the model against the ice, and the second method is to move the ice sheet against the model. For ice ridge testing, the method used at HSVA is to pull the model using a carriage. In case of offshore structure the method employed depends on the project requirements.

The construction of ice ridges in basins are experience-based and lack of standardization for ice ridge creation. Fenz et al. (2016) discussed the influence of the ice ridge preparation in the physical properties of the ridge and concluded that the temperature and the block size can

modify the keel cohesion. Høyland et al. (2001) discussed the techniques applied at HSVA in order to guarantee the scale ice properties. The main parameters are the thickness of the surrounding level ice and the strength of the consolidated layer where the temperature is associated to the strength of the ice, cold temperature leads to higher ice strength.

In this scenario, the development of computational simulation aims to optimize the model scale test where the designer would select a variety of hulls and ice ridge geometries to simulate in the software. Furthermore, the physical properties of the ice ridge can be modified in order to evaluate their influence on the loads. Full-scale ice ridges have a wide range of strength, and the numerical simulation allows the user input the desired value, while model testing requires more effort to keep the scaling properties.

Nonetheless, the software does not substitute the experimental ice ridge testing because it does not simulate the propeller and ice interaction in the case of ship testing, which may have significant influence during astern breaking of ice ridges.

1.3 Ship and Ice Ridge Interaction

To determine the vessel capability to overcome the ice ridge, the sum of kinetic energy and the propulsion energy available for the vessel needs to be higher than the energy to break the ice ridge. When the ship penetrates the ice ridge, the velocity decreases due to the usage of the kinetic energy to break the ice ridge. In some cases, the ship comes to a complete stop. Ehle (2012) described the fact that a ship either passes through the ice ridge at once or by a series of trials called ramming.

The ice resistance depends on the profile geometry of the ridge, the ship hull’s geometry, the rubble ice geometry and internal cohesion. In addition, the velocity in which the ship reaches the ridge has a direct influence on the capability to overcome the ridge. Figure 4 shows the phases in which the vessel has to face when passing through the ice ridge. The plot shows that the ice ridge resistance depends on the ship’s position in relation to the ice ridge.

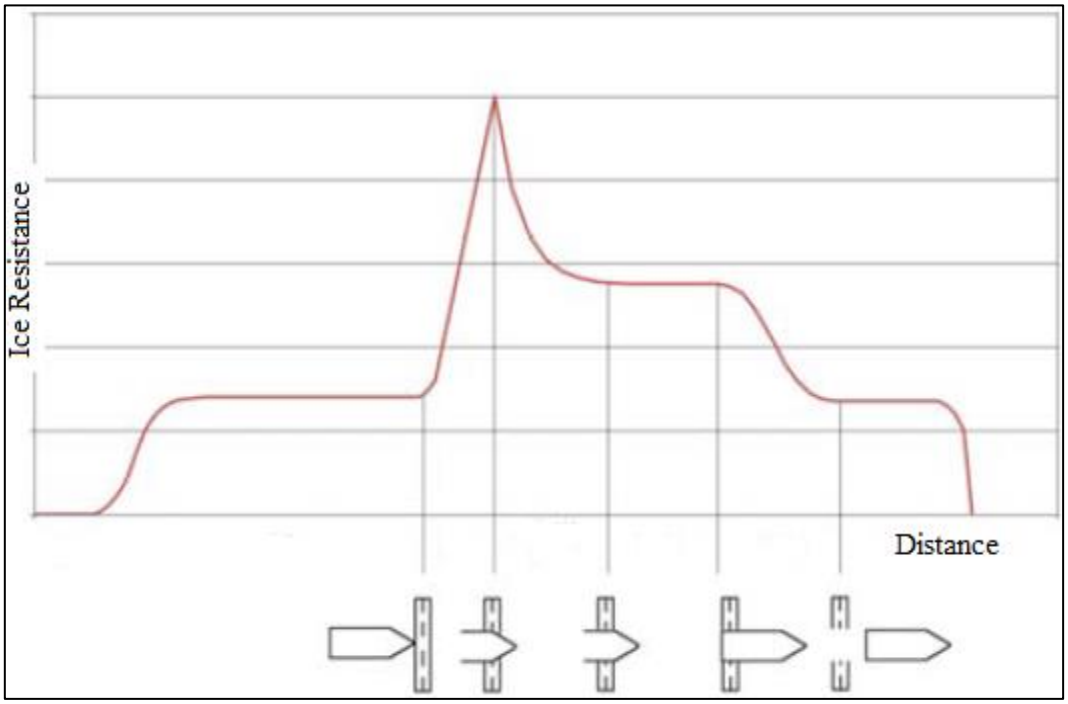

Figure 4. Breaking through ice ridges. Adapted from Ehle (2012)

Figure 5 shows an example of ship and ice ridge model scale at the HSVA ice basin. The ice ridge sail is the only part visible from outside of the basin. The tank is surrounded by level ice and the self-propelled model is controlled to reach the ice ridge.

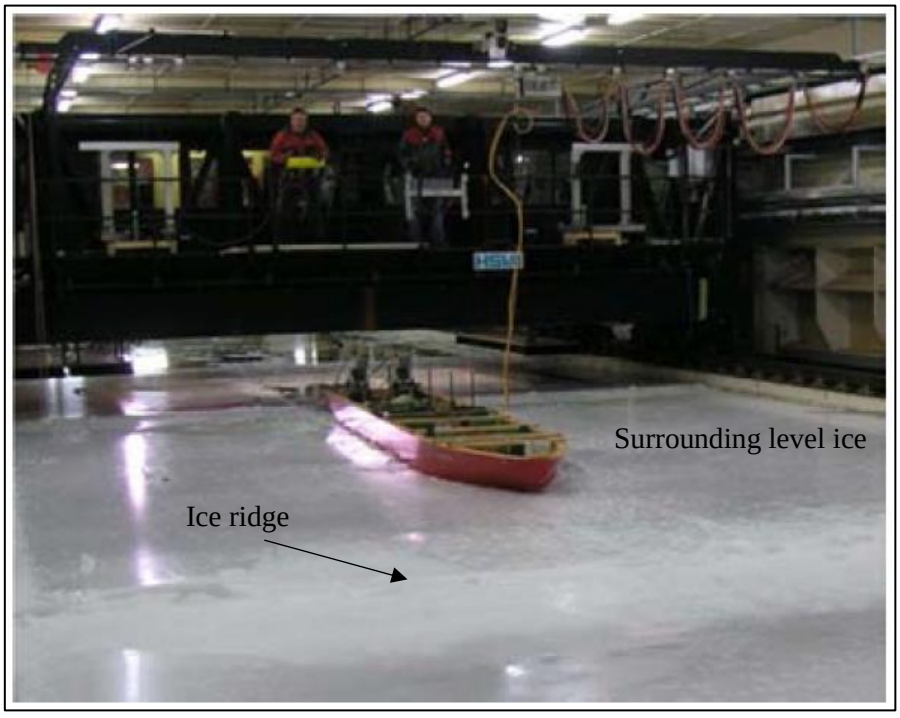

Figure 5. Ship simulation in ice tank. Adapted from Ehle (2012)

1.4 Punch Test

The punch test is a standardized method to determine the internal shear strength of the rubble ice accumulated in the ridge keel in the ice ridge. The consolidated layer is removed for the test so that the measured forces are only due to the interaction between punch device and unconsolidated layer.

The tests consist of a cylinder lowered through the ice ridge with constant vertical velocity of 6.7 mm/s. There are two cylinders available to perform the test; one cylinder has 180 mm in diameter and weighs 88.5 kg while the other device has 250 mm in diameter and weighs 122 kg.

Figure 6 shows that it is possible to identify the rubble ice accumulated under the punch device after transposing the ice ridge.

Figure 6. Punch test at HSVA ice basin

1.5 Numerical Simulation

The software under development at HSVA simulates the ice ridge creation, the ship simulation and the punch test. The software outputs the forces, velocities, acceleration and position of the structure under analysis. In addition, the code exports *vtk* format files of the geometry. To analyse a structure and ice ridge interaction the qualitative and quantitative results are necessary to draw conclusion from the simulation.

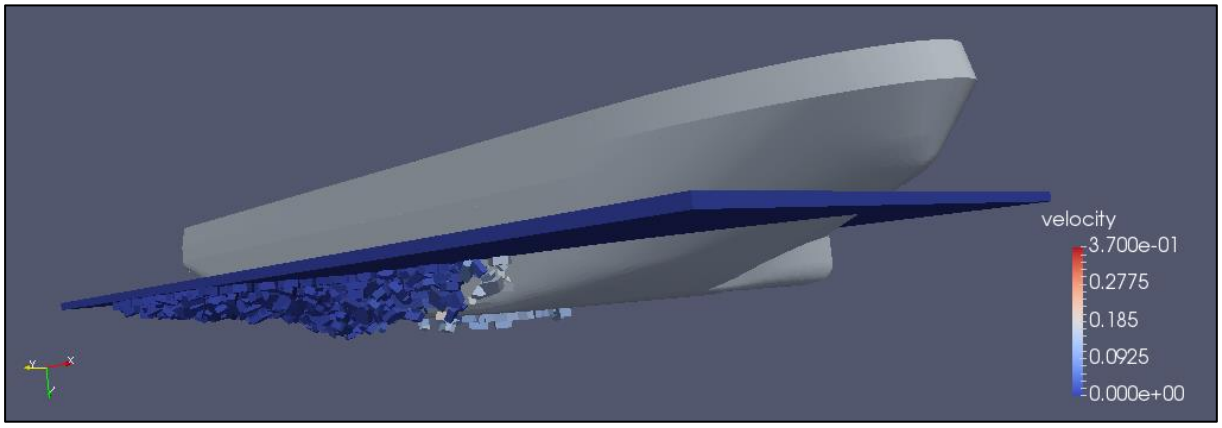

Figure 7. Example of a simulation of ship and ice ridge interaction

2. DISCRETE ELEMENTS METHOD

2.1 Numerical Scheme

The numerical scheme for the discretization of the equations in the software is the discrete elements method (DEM). DEM is capable to simulate rotational dynamics of complex geometries by addressing the problems involving discontinuous or granular materials. DEM is largely applied to mining engineering and is recently adapted to Arctic engineering.

Other researchers used different numerical schemes for ice ridge simulation. For instance, Serré (2011) used Finite Elements Methods (FEM) to simulate punch tests. Polojärvi and Tuhkuri (2008) performed three-dimensional punch tests combining FEM and DEM. Ji et al. (2014) simulated a conical structure using DEM.

DEM simulates the motion and interaction of large number of particles in a macroscopic scale. The DEM method used in the software under development at HSVA is derived from Matuttis and Chen (2014), who elaborated a DEM algorithm for three-dimensional elements where the contacts between elements allow the calculation of the forces involved. In addition, the Young's modulus of the material is necessary to compute the elastic forces from the overlap between particles because they are not rigid.

When two elements are in contact, an overlap region is formed as shown in Figure 8. Point *P* in Figure 8*b* represents the centroid of the overlap geometry and the point *C* represents the centre of gravity of the particle. The direction of the normal force is the weighted averaged between the vertices S_1 and S_2 to *P*, where the intersection of the boundaries of the particles creates the vertices S_1 and S_2. The force magnitude depends on the contact area for two-dimensional problems and the intersection volume for three-dimensional problems.

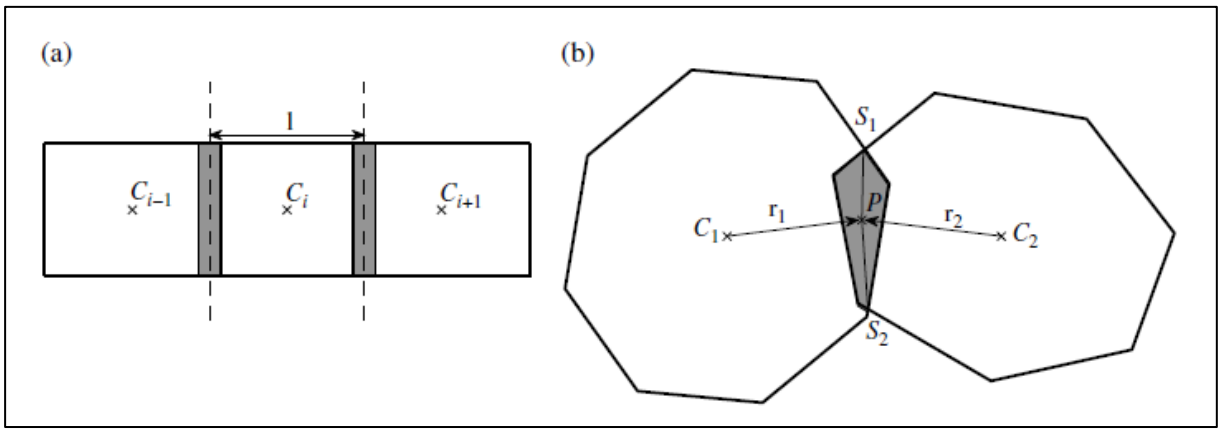

Figure 8. Contact between two elements. Adapted from Matuttis and Chen (2014).

According to Matuttis and Chen (2014), the particle's shape has an influence on the dynamic calculation. The elastic force for rectangular shape geometry is linearly proportional to the overlap's depth δ, for spherical contact the force is proportional to $\delta^{3/2}$, and wedge-shaped contacts is in relation to δ^2. Figure 9 illustrate this concept. For an accurate calculation, the geometry representation should be as realistic as possible.

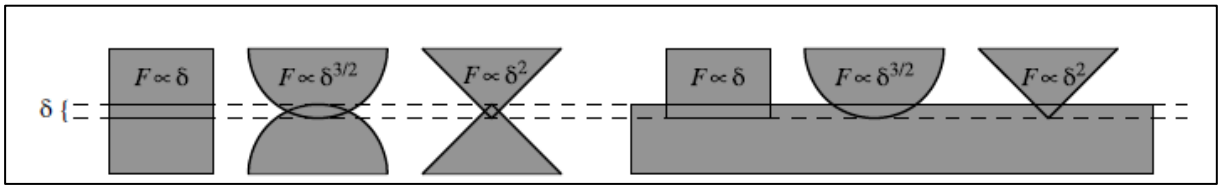

Figure 9. Particle deformation. Adapted from Matuttis and Chen (2014).

2.2 Contact Detection

Seidel (2016) introduced in the software a contact detection algorithm between elements. Before calculating the forces, the software has to identify the pairs of elements that are in contact among all of them in the simulation. For algorithm efficiency, the code identifies the particles that are in contact using the *sweep and prune* algorithm. Based on the elements axis-aligned bounding box, the code searches those elements nearby and identify the intersections.

The elements in DEM are described by a triangular mesh as shown in Figure 10. The triangulation is necessary to simplify the data structure of the algorithm for elements contact detection.

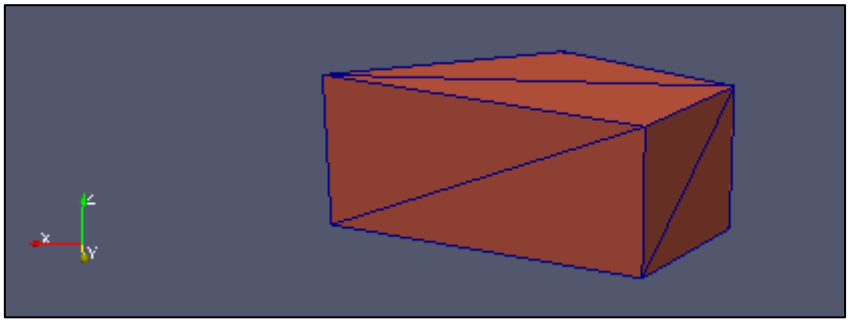

Figure 10. DEM particle with triangle mesh

To determine the overlap between two elements it is necessary to compute the overlap of the faces. For this task, a triangle intersection check is performed by using the point-normal form. The algorithm calculates the scalar parameter λ, which contains the position of a point along a line. The equation of a line *r* passing through the three-dimensional points V_1 and V_2,

where the edge of the triangle is defined as (V_2 - V_1) is written in the point-parameter form. The intersection point of a line with a plane is *d* and *n* is the normal vector of the plane.

$$n \cdot r = d \tag{2}$$

$$r = V_1 + \lambda(V_2 - V_1) \tag{3}$$

$$\lambda = \frac{d - n \cdot V_1}{n \cdot (V_2 - V_1)} \tag{4}$$

The possible solutions for λ are illustrated in Figure 11 and discussed below:

- If $0 < \lambda < 1$, there is an intersection between the triangles. Cases *d* and *e*.
- If $\lambda = 0$ or $\lambda = 1$, one of the triangle's vertices is on the plane. Cases *b*, *c* and *d*.
- If $\lambda > 1$ or $\lambda < 0$, there is no intersection. Case *a*.
- If there is no solution, one of the vertices V_1 and V_2 is on the plane or parallel to it. Cases *c* and *f*.

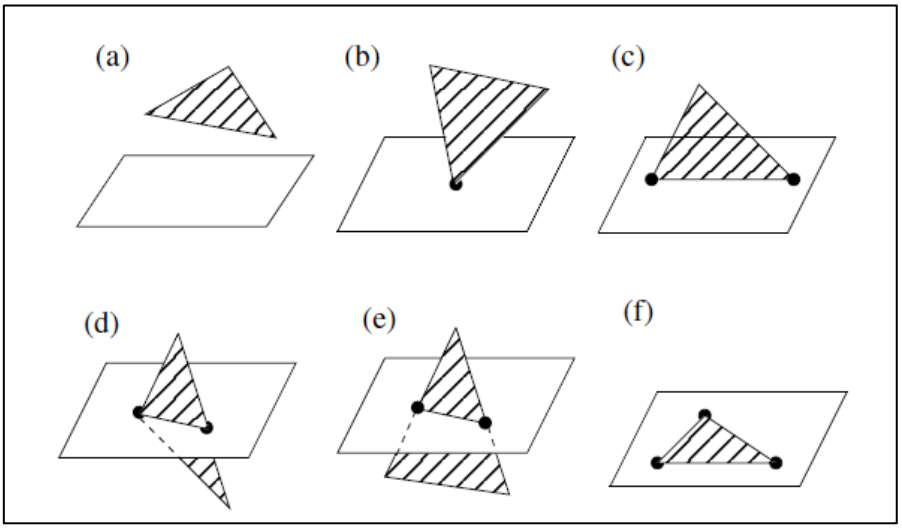

Figure 11. Triangle and plane intersections. Adapted from Matuttis and Chen (2014)

After identifying the triangle contact, the algorithm calculates the volume, centroid of the overlap, superficial area of the overlap region and force direction based on the polygon created from the overlap of two elements P_1 and P_2 as illustrated in Figure 12.

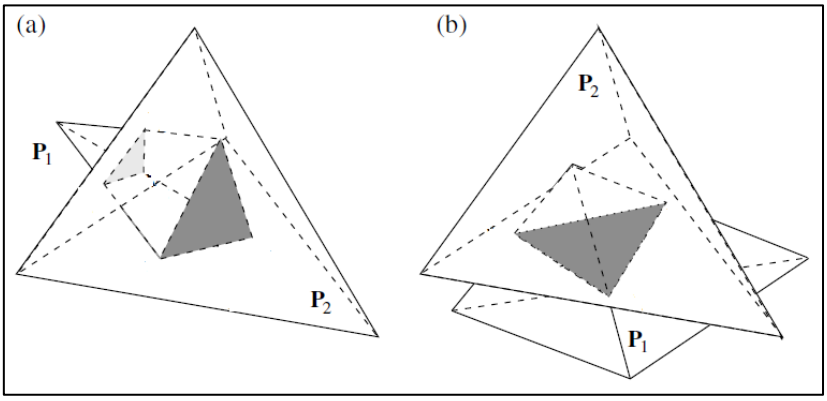

Figure 12. Overlap geometry. Adapted from Matuttis and Chen (2014)

2.3 Forces

The force calculation is performed for all the pairs of elements that are in the list of intersection pairs obtained from the contact detection. The forces between two elements are the sum of normal F_n, tangential F_t, buoyancy F_b, gravity F_g and drag forces F_d.

$$\overrightarrow{F_{total}} = \overrightarrow{F_n} + \overrightarrow{F_t} + \overrightarrow{F_b} + \overrightarrow{F_g} + \overrightarrow{F_d} \tag{5}$$

2.3.1 Normal Force

The normal force is calculated as the sum of the elastic force F_e, the damping force F_{dn} and cohesive forces F_{coh}. The normal vector is n.

$$\overrightarrow{F_n} = (\overrightarrow{F_e} + \overrightarrow{F_{dn}} + \overrightarrow{F_{coh}}).\vec{n} \tag{6}$$

The elastic force is computed based on the smallest Young's modulus Y between the two elements in contact and a characteristic length L_c as described in Johannes (2016). The overlap volume V_0 is obtained from the contact detection algorithm.

$$\overrightarrow{F_e} = \frac{Y \cdot \overrightarrow{V_0}}{L_c} \tag{7}$$

The damping force is calculated based on the time derivative of the overlap volume. It also depends on the reduced mass m_{red} calculated from the mass of the particles in contact.

$$\overrightarrow{F_{dn}} = \gamma_n \sqrt{\frac{Y \cdot m_{red}}{L_c^3}} \cdot \frac{\overrightarrow{\delta V_0}}{\delta t} \tag{8}$$

Normal cohesive forces F_{coh} only applies to the interaction between two rubble ice particles. It depends on the cohesive Young's modulus Y_{coh} and the contact area A_0.

$$\overrightarrow{F_{coh}} = Y_{coh} \cdot \overrightarrow{A_0} \tag{9}$$

2.3.2 Tangential Force

The tangential force is the sum of the friction F_{fc} and dissipative forces F_{dt}.

$$\overrightarrow{F_t} = \overrightarrow{F_{fc}} + \overrightarrow{F_{dt}} \tag{10}$$

According to Matuttis and Chen (2014) there is no exact analytical method for many-particle friction, being necessary to model the phenomena. Knowing relative velocity and contact point between elements is sufficient to use the Cundall-Strack model. The model calculates the projection of the tangential force f^p from time step t-τ onto the tangential plane of the time step t during the advance from time step t-τ to time t.

$$\overrightarrow{f^p_{(t-\tau)}} = \overrightarrow{f_{(t-\tau)}} - (\overrightarrow{f_{(t-\tau)}} \cdot \overrightarrow{n_t})\overrightarrow{n_t} \tag{11}$$

The magnitude of the force at time step t-τ is rescaled f^r.

$$\overrightarrow{f^r_{(t-\tau)}} = \left|\overrightarrow{f_{(t-\tau)}}\right| \cdot \frac{\overrightarrow{f^p_{(t-\tau)}}}{\left|\overrightarrow{f^p_{(t-\tau)}}\right|} \tag{12}$$

The dissipative force in the tangential direction is dependent of the dissipation coefficient γ_t, and the tangential velocity v_t.

$$\overrightarrow{F_{dt}} = -\gamma_t \sqrt{Y_t \cdot m_{red} \cdot L_c} \cdot \overrightarrow{v_t} \tag{13}$$

2.3.3 Buoyancy and Gravity Forces

The buoyancy force takes into account the submerged volume *V* of the element and water density ρ_w

$$\overrightarrow{F_b} = \rho_w V \vec{g} \tag{14}$$

The gravity force depends on the mass of the element *m*

$$\overrightarrow{F_g} = m\vec{g} \tag{15}$$

2.3.4 Drag Force

The rubble ice elements are totally submerged, therefore the particles are subjected to the resistance from the water when moving. The drag force calculation is described as:

$$\overrightarrow{F_d} = \frac{1}{2}\rho_w \overrightarrow{v^2} C_d A_{av} \tag{16}$$

Where ρ_w is the water density, *v* is the element velocity, C_d is the drag coefficient and A_{av} is the average superficial area. Due to rubble ice rotation, the frontal area might change at each time step. To optimize the calculation, an average superficial area is taken for the calculation of the drag force.

2.4 Torque

To obtain the torque, the point where the force is applied and the distance to the centre of mass of the element are known. Therefore, the torques can be determined based on the vectors r_1 and r_2 representing the distance from the centre of mass of element 1 and 2 respectively to the point of application of the force.

$$T_1 = r_1 \times f \tag{17}$$

$$T_2 = r_2 \times (-f) \tag{18}$$

2.5 Forces on Ship

Alekseev (2016) introduced the algorithm for the calculation of the ship buoyancy in the simulation. Due to the changes in the position of the ship during the simulation, an update buoyancy value at each time step is required.

Thus to calculate the buoyancy of the ship, the Simpson's first rule is used. The submerged part of the ship hull is divided into equally spaced cross sections. The integration of these areas provided the underwater volume of the ship.

Thrust is assigned to the ship in the simulation based on the ship's propeller curve. The thrust is obtained from the non-dimensional parameters of the propeller: advance ratio, thrust coefficient, torque coefficient and open water efficiency.

2.5.1 Ice Resistance of Ships

The ice resistance due to the surrounding level ice is simulated based on analytical formulation developed by Lindqvist (1989); there is no graphical output of the level ice breaking. The resistance of ship going through ice is based on the main dimension of the ship, hull form, ice thickness, friction and ice strength. The empirical resistance curve for open water of the ship is included in Lindqvist formulation.

The ice resistance R_{ice} is dependent on the crushing force $R_{c,}$ bending force $R_{b,}$ the velocity of the ship v, the ice thickness H_{ice} and the length of the ship L. In Lindqvist (1989), there is further details on the calculation of R_c, R_b and R_s.

$$R_{ice} = (R_c + R_b)\left(1 + \frac{1.4v}{\sqrt{g.H_{ice}}}\right) + R_s\left(1 + \frac{9.4v}{\sqrt{g.L}}\right) \tag{19}$$

3. RUBBLE ICE GENERATION

The rubble ice in the software is modeled as a discontinuous media with well-defined geometrical and mechanical properties. The geometrical properties of the rubble have not been extensively researched and there is a lack of information in regards to the ice particle shape in nature and model scale. This thesis investigates the geometry of model scale rubble ice from ice ridges generated at the HSVA ice basin.

The first version of the software developed by Seidel (2016) and Alekseev (2016) used rectangular shaped ice rubble elements. Initially, the user had to input the length, width and thickness of the rubble ice and the algorithm applied a random variation of 50%, 10% and 5% respectively when creating the particles in the simulation. This approximation is reasonable; however, the geometrical shape of rubble ice is closer to polygons than rectangles. In addition, the variability of the particle's main dimensions should be controlled to allow repeatability of the simulation.

In order to extract the particle geometry information from model scale and implement into the algorithm, measurements of rubble ice particles from model ice ridge were performed.

3.1 Rubble Ice Measurement Methodology

To introduce polygon shaped particles into the code, rubble ice from 24 ice ridges constructed at HSVA were measured. After a series of tests, each ridge was dismantled by pushing the rubble ice out of the ridge so that it floats on the ice basin water. A total of 33 images were processed and analysed. In 21 images, samples of rubble ice were taken from the water and put outside the basin as shown in the bottom picture of Figure 13. The remaining 12 pictures were taken directly from the water without choosing samples as shown in the top picture of Figure 13.

HSVA had the images of the rubble ice in its database and the work performed was the image processing. The researcher processed and analysed those images using ImageJ open software version IJ 1.46r.

3.2 Image Processing and Analysis

Step 1: The image was transformed into black and white, and the brightness of the image is adjusted for better contrast between particles and background. Some of the pictures had uniform lighting and this adjustment did not affect the quantity of particles visualized in the picture after this process, as seen on the bottom of the Figure 13 and Figure 14.

However, the pictures taken directly from the water, Figure 13 (top) and Figure 14 (top) had a non-uniform light distribution. Hence, after the brightness adjustment, the particles located in the centre of the image did not appear in the picture. If the brightness of the image was decreased, the boundary between the particles could not be identified.

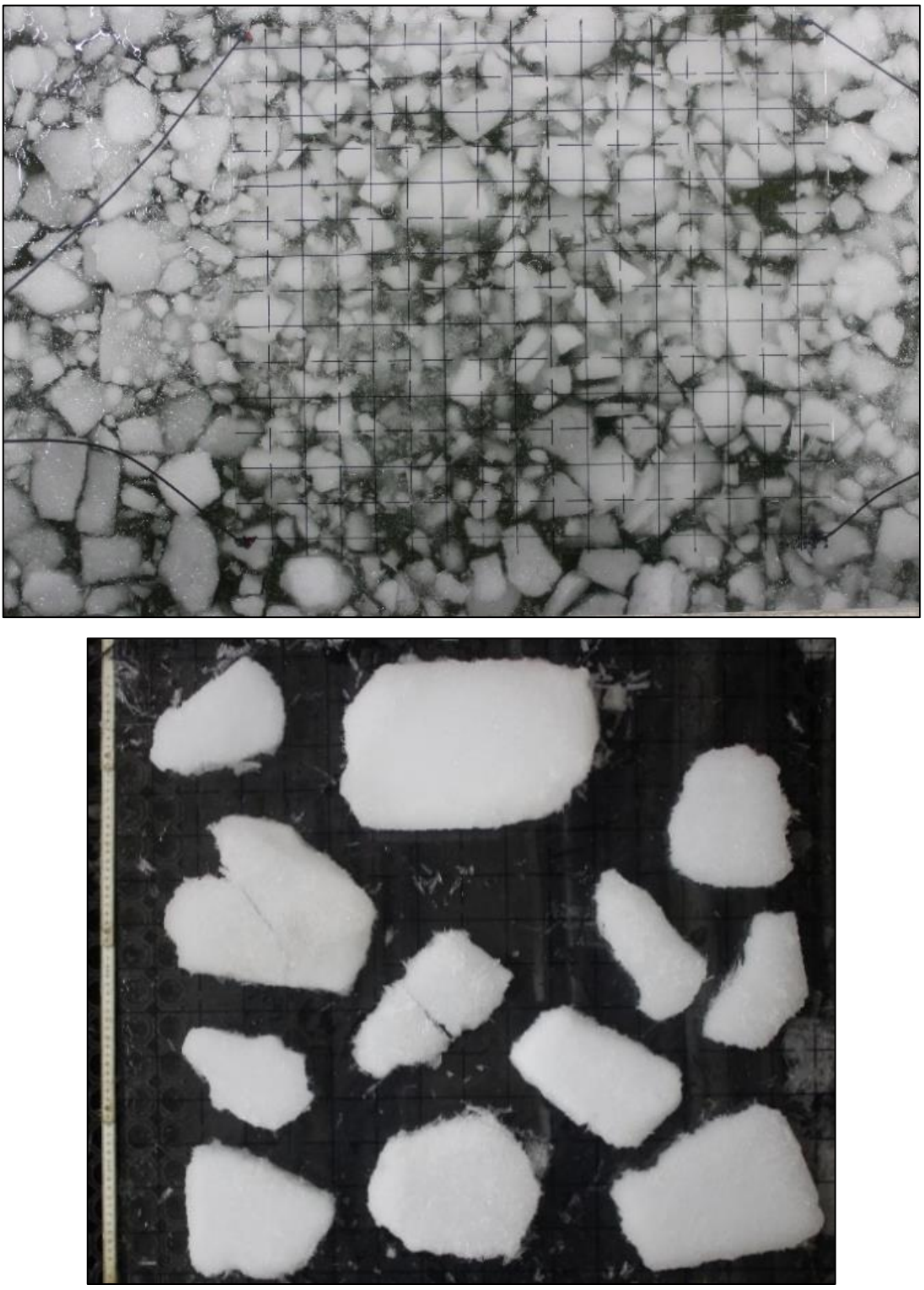

Figure 13. Picture at the ice tank. On the water (top), outside the basin (bottom)

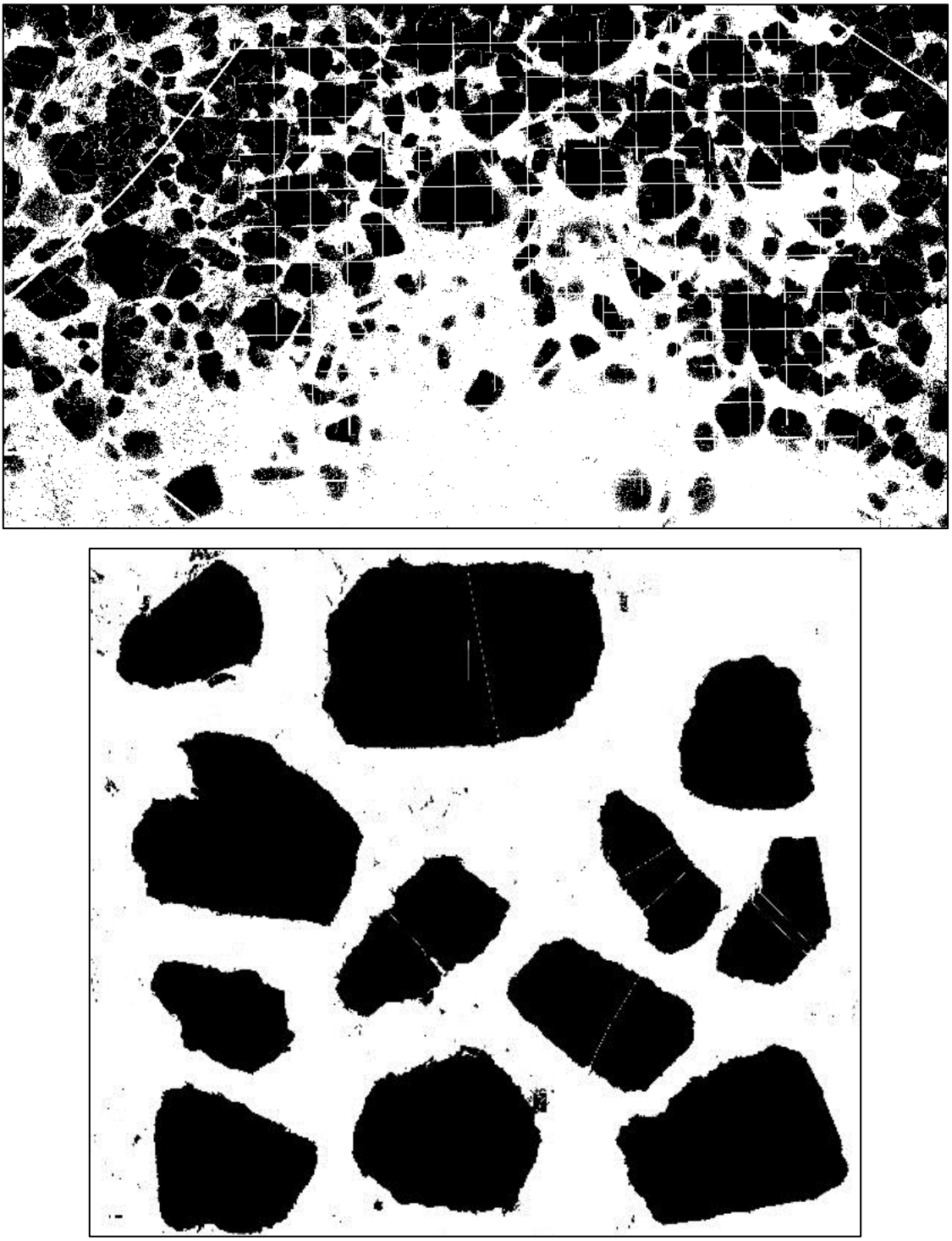

Figure 14. Pre-processed image. On the water (top), outside the basin (bottom)

Step 2: The boundaries of each particle was identified by applying the tool *watershed* developed by Roerdink and Meijster (2001). This algorithm execute the image segmentation by isolating objects from the background. The aim of this tool in the analysis is to ensure that the particles do not touch one another by identifying its edges.

The algorithm to perform boundary identification was design for 8-bit image; therefore, the conversion from RGB color to 8-bit is necessary. A pixel stores the intensity varying from black to white changing its intensity and the level of precision. A 2-bit image has 2^2 tones: 00 (black), 01 (gray), 10 (gray), 11(white), a 4-bit image has 2^4 tones and 8-bit has 2^8 tones.

Step 3: After editing the images, a scale was set based on the distance between two pixels in the image, usually with the help of a ruler that is in the picture. The particles are identified based on its superficial area. To reduce the processing time by excluding worthlessly small elements, a minimum superficial area is set. Any particle with superficial area below this limit is neglected by the image analysis. Even though some ice particles are in the picture, they are not necessarily rubble ice from the ridge. Such small pieces are originated from broken level ice or broken rubble ice due to the manipulation of the ice ridge. The minimum area for the analysis is set to 10% of the target dimension of the desired rubble ice.

The post-processed image is shown in Figure 15 where each rubble ice identified has an ID number. The summary of the image processing methodology is described in the Table 1.

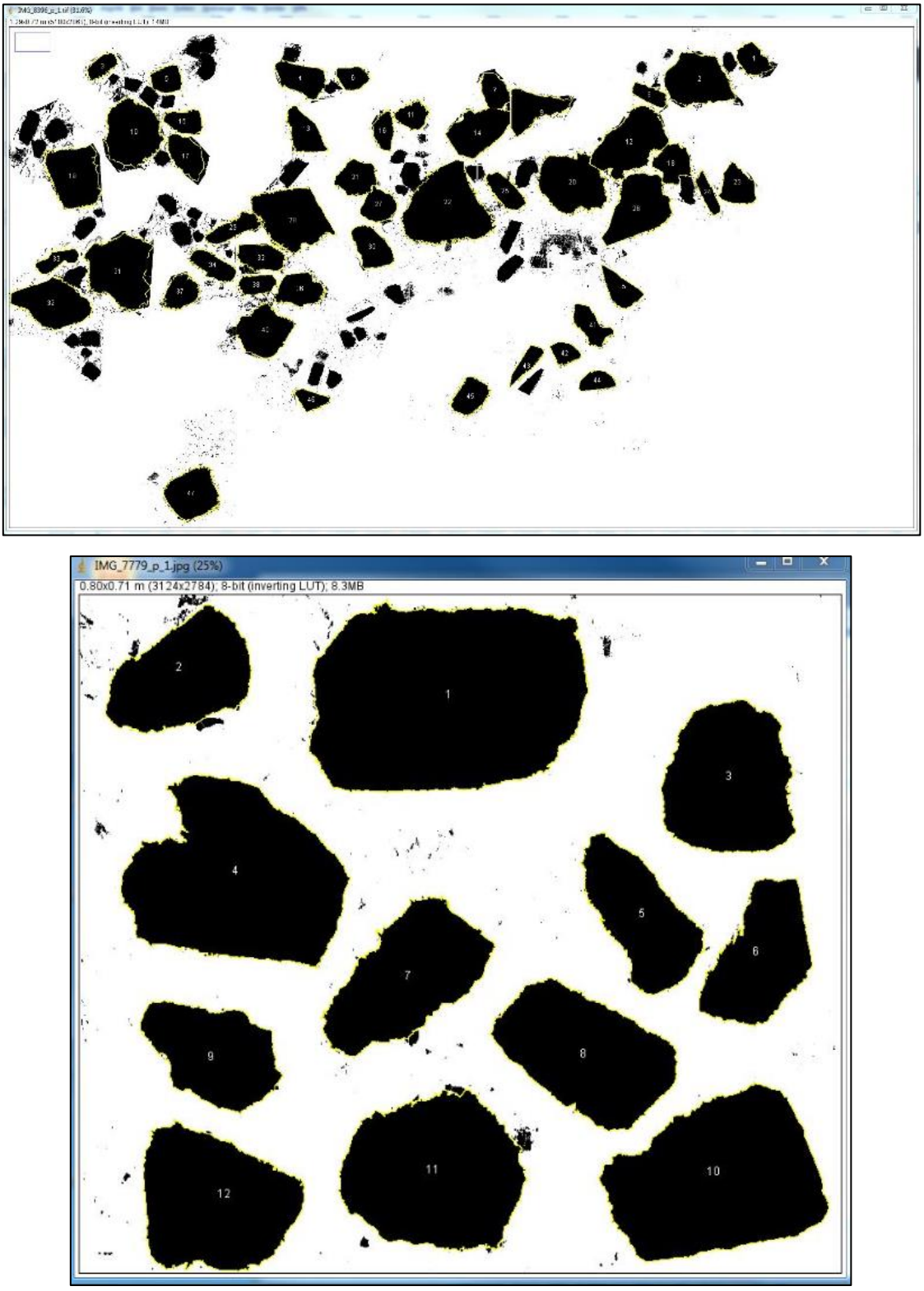

Figure 15. Post-processed image. On the water (top), outside the basin (bottom)

Table 1. Image processing summary

Total of pictures analysed	33 *jpeg* images
Ice pieces identified	1225 rubble ice
Target area	100 mm x 100 mm
Minimum area for particle detection	30 mm x 30 mm

The image analysis outputs the width and height of the axis-aligned bounding box as described in Figure 16. This information is introduced in the code through statistical analysis discussed in section 3.3.

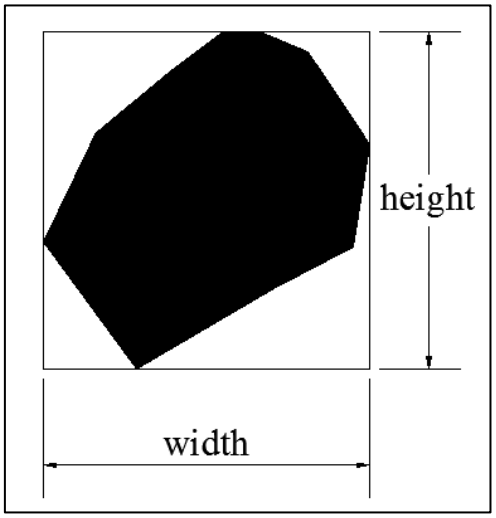

Figure 16. Example of ice particle axis-aligned bounding box

3.3 Rubble Ice Dimension Analysis

The results of the measurements of the rubble ice particles described in section 3.2 are plotted in histograms. From the histogram data, a probabilistic density function (PDF) is obtained, which describes the relative probability for a random variable to stay within a range.

A variety of models for the probabilistic distribution function is suitable for this data analysis. Three PDF curves were compared: gamma, beta and Weibull distribution. The shape of the curve of those distributions is defined by two parameters. One is the characteristic size value and the second parameter is the width of the distribution.

The histograms are processed in MATLAB using the *histfit* function. The function fits the PDF curve according to the histogram inserted and outputs the two shape parameters that define the curve. Figure 17 and Figure 18 represent the width and height of the particle's axis-aligned bounding box respectively. The *x*-axis represents the dimension measured and the *y*-

axis the probability density function of each dimension. The area underneath the curve represents the probability of a given range.

The Weibull distribution is usually applied to time related problems, therefore not utilized in this problem. Gamma and beta are mainly used for statistical description of positive and real numbers. Both distribution curves presented similar results for the data analysed. The difference among them is that the beta distribution is designed only for parameters ranging from [0, 1]. In addition, the algorithm for beta distribution is easier to implement than the gamma distribution. Therefore, the beta probability density function is chosen and it is described by the equation 20.

$$f(x) = \frac{1}{B(a,b)} x^{a-1}(1-x)^{b-1} \tag{20}$$

Where *f(x)* is the probability density function, *a* and *b* are the positive shape parameters and *B(a,b)* is the beta function. The values of the shape parameters from the measurements are summarized in Table 2 and Table 3.

Table 2. Shape parameters for height of axis-aligned bounding box

a	6.47
b	73.89

Table 3. Shape parameters for width of axis-aligned bounding box

a	6.22
b	72.09

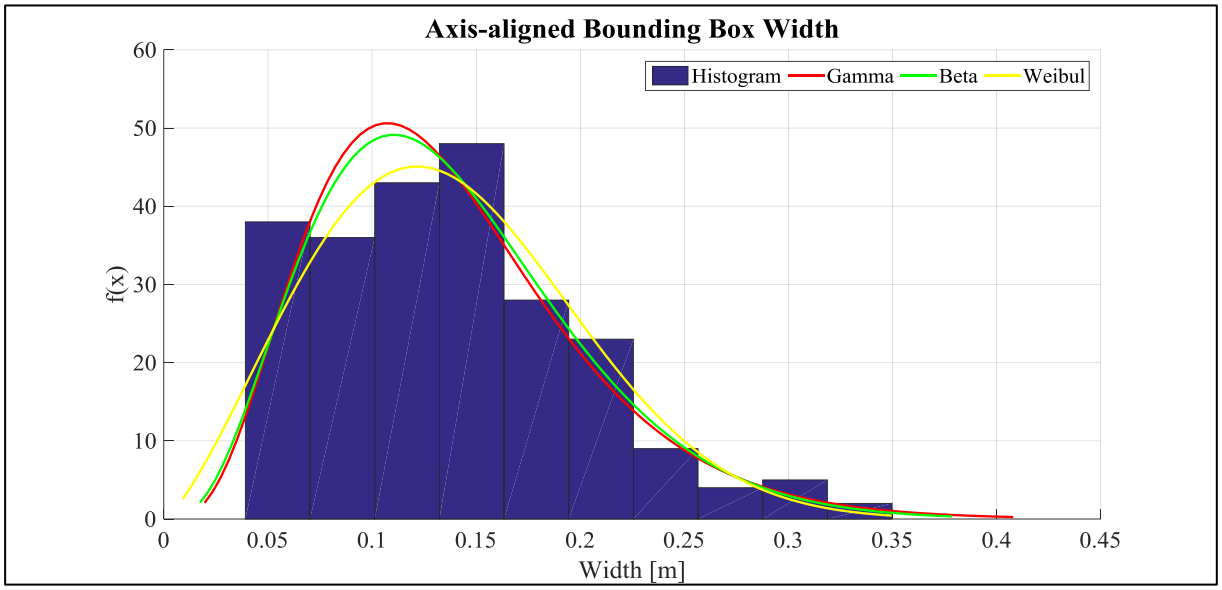

Figure 17. Histogram from width of the axis-aligned bounding box measurements

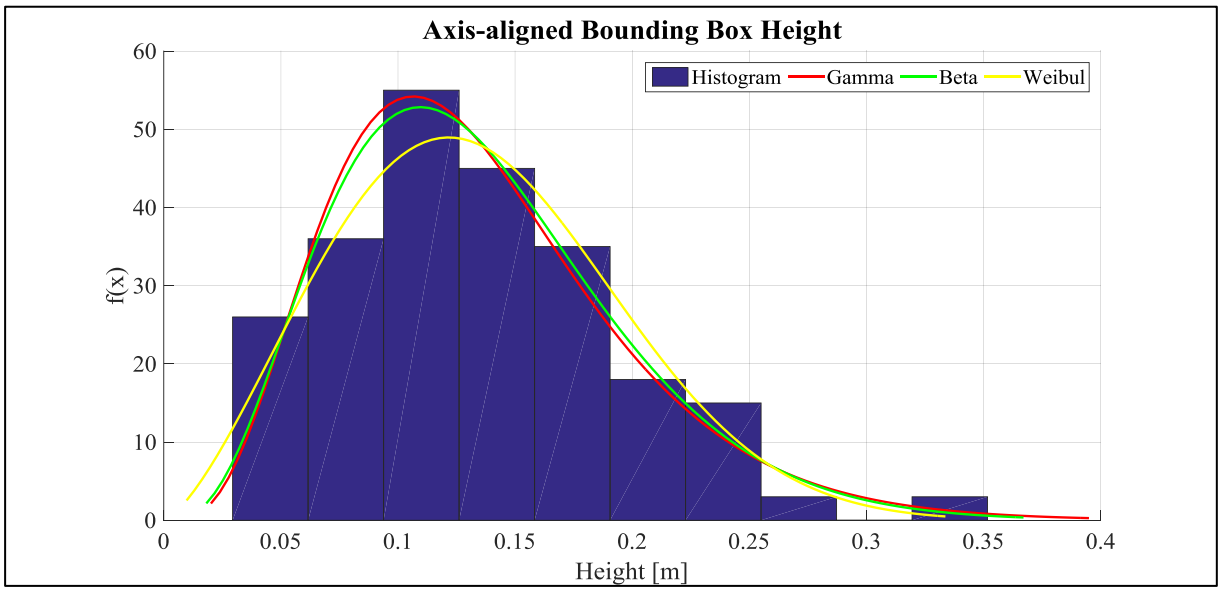

Figure 18. Histogram from height of the axis-aligned bounding box measurements

Another parameter from the rubble ice input into the code is the number of edges of a particle. The software ImageJ does not provide this information, therefore the number of edges for each particle was collected by visual inspection of the images. The result of this process is expressed in the Figure 19. Since the beta distribution deals with data ranging from [0, 1], the number of edges was divided by 10 because there were no particles with number of edges larger than that.

Table 4. Shape parameters for number of edges

a	14.16
b	12.23

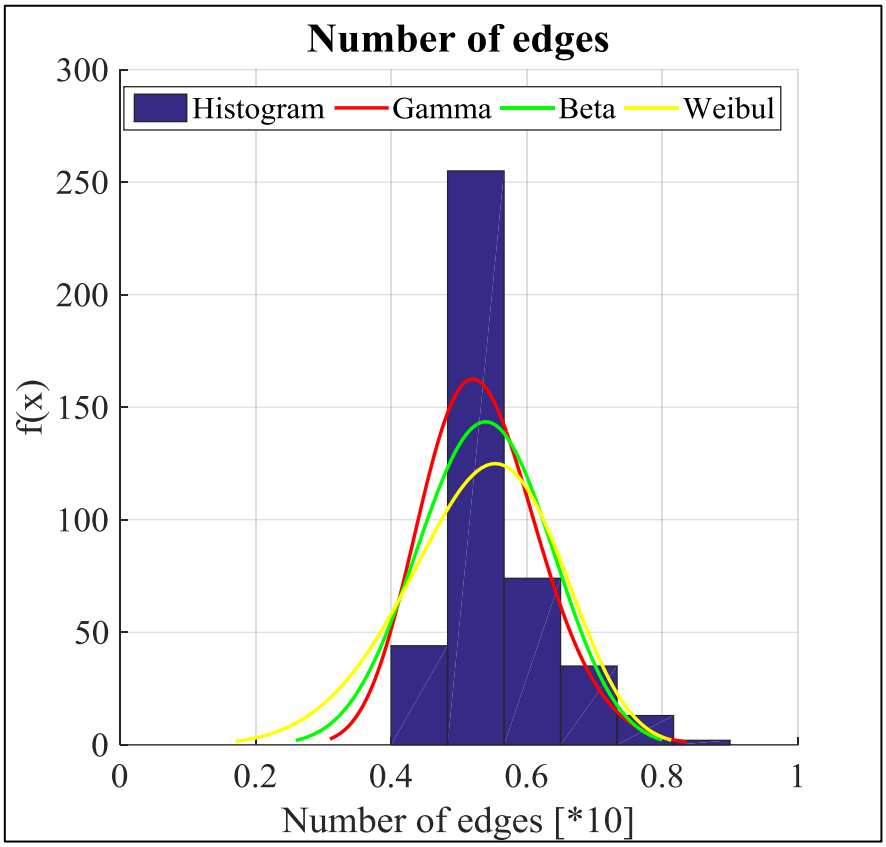

Figure 19. Histogram from number of edges measurements

3.4 Rubble Ice Geometry Algorithm

3.4.1 Random Generator

In order to input the experimental data into the software, an algorithm is used to generate random number of edges, width and height of the particles based on shape parameters of the beta distribution curves obtained from the measurements.

The FORTRAN algorithm developed by Brown and Lovato (2013) is introduced in the code, which uses the numerical method described by Cheng (1978). The number generator is based on rejection method, that is, the candidate numbers U_1 and U_2 are randomly generated and subjected to trials of rejection or acceptance. The method covers positive shape parameters $a, b > 0$. The algorithm to generate the particle geometry runs in the pre-processing phase of the program as described in Figure 20.

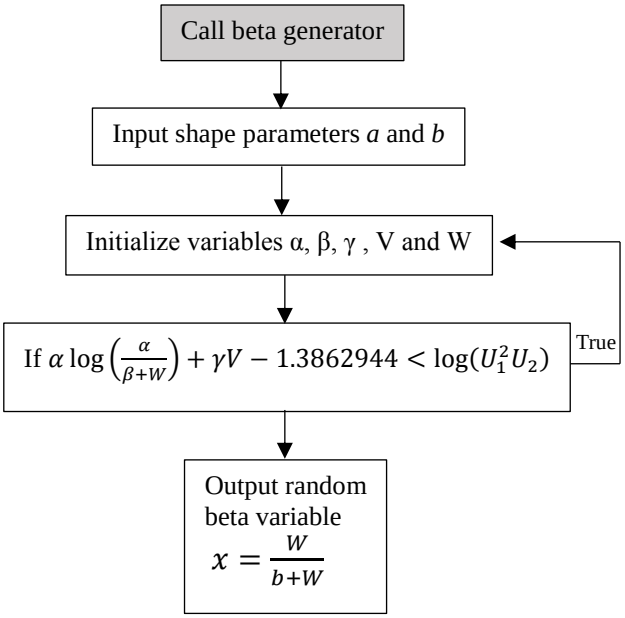

Figure 20. Random beta PDF algorithm

To verify the results of the random generator, tests were performed by running the code to generate the same amount of particles as obtained from the image processing. Figure 21, Figure 22 and Figure 23 compare the width, height and number of edges respectively, the samples generated by the algorithm and the data experimental data collected from the experiment.

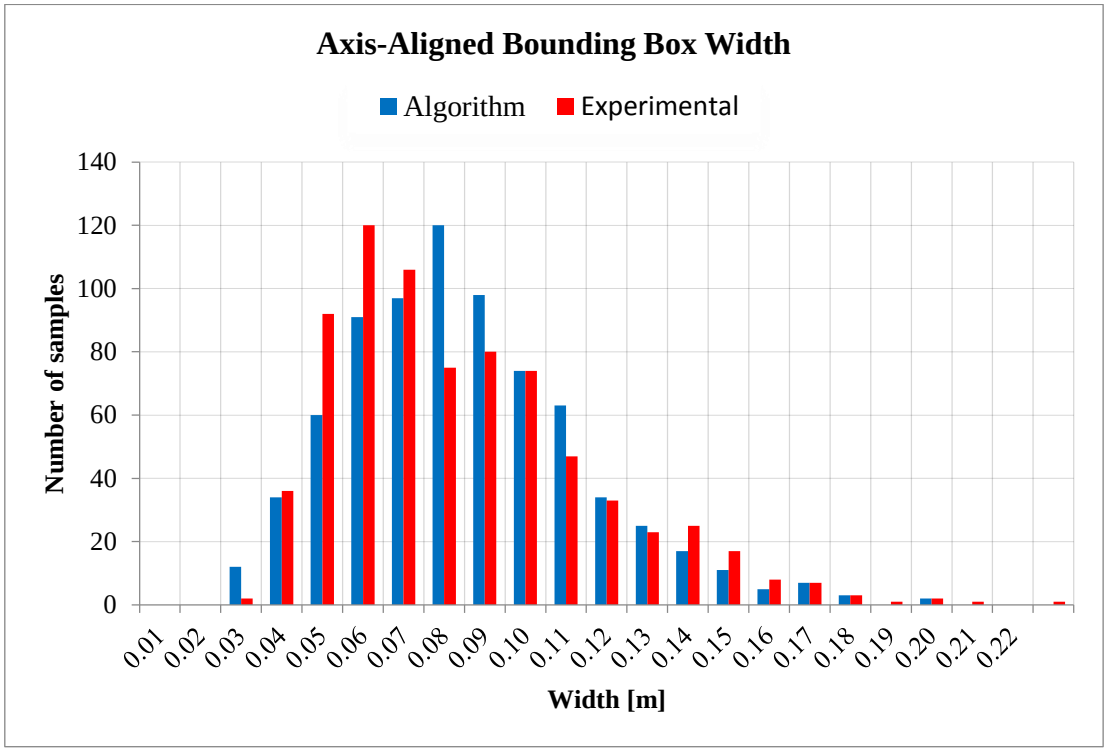

Figure 21. Histogram: Axis-aligned bounding box width

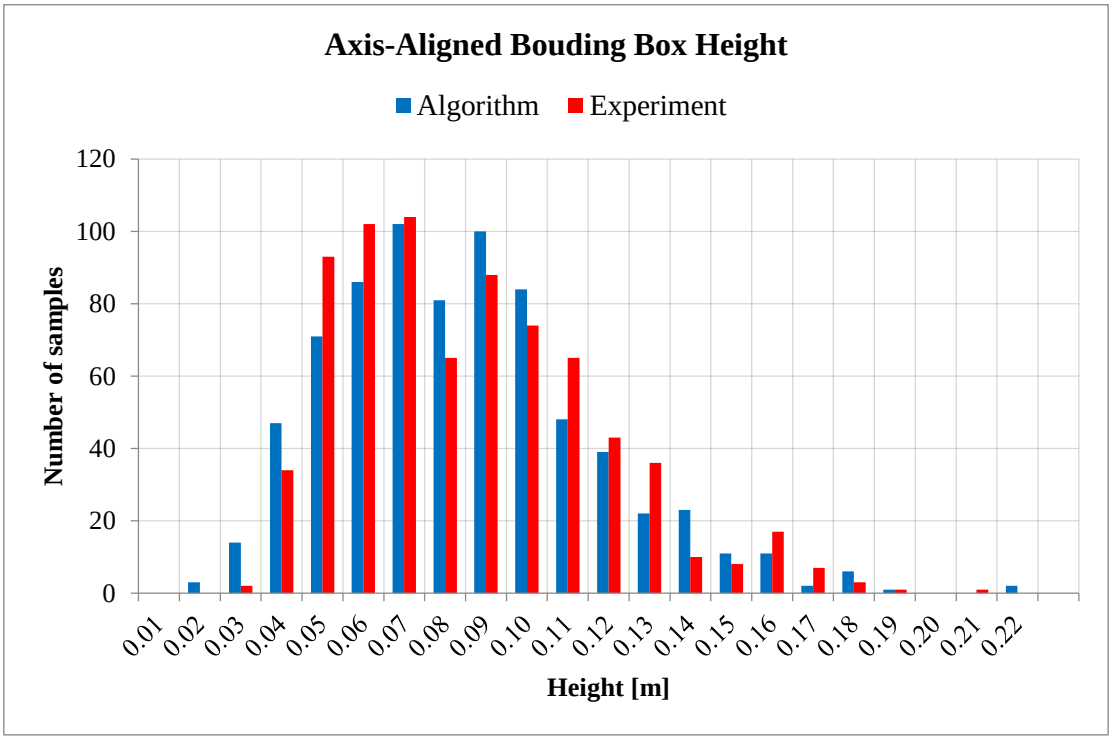

Figure 22. Histogram: Axis-aligned bounding box height

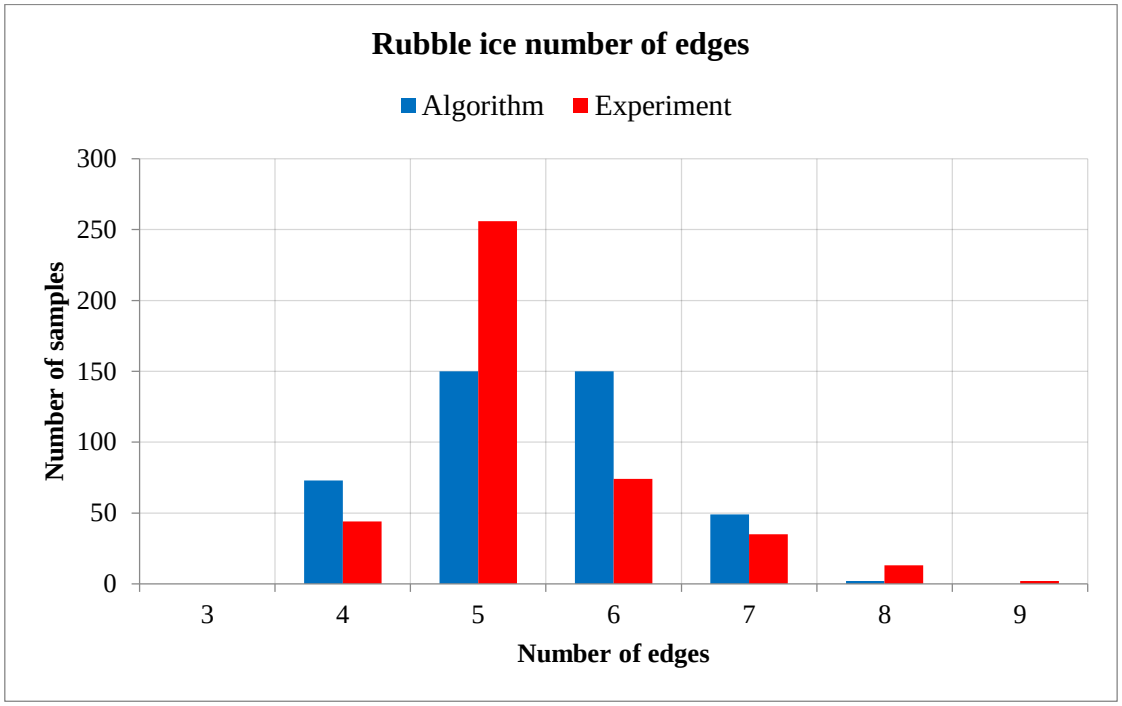

Figure 23. Histogram: number of edges

The algorithm to generate the number of edges has to have a controlled output range because of the geometrical algorithm described in section 3.4.2. Since the parameter is obtained from the probabilistic distribution, there is the possibility to obtain extreme values, such as number of edges equal to 2 or 10. In order to simplify the algorithm, the output of number of edges from the random generator is within the range [4, 8] edges. The analysis of the experimental data showed that 97% of the number of edges for the rubble ice is inside the mentioned range. Therefore, if the number of edges obtained from the random generator is less than 4 edges, it becomes 4. If the number of edges is more than 8, it becomes 8. Figure 23 compares the experimental data and the algorithm.

The median values of the plots are compared in Table 5 showing that the algorithm is suitable to represent the experimental data.

Table 5. Comparison between algorithm and experimental data median values

	Width [m]	Height [m]	Number of edges
Experimental data	0.080	0.081	5.35
Algorithm	0.079	0.079	5.43
Difference	0.69%	1.45%	-1.50%

The height and width dimensions are almost identical and the aspect ratio of the particle is near to 1. All the analysis executed did not included the ice thickness, which is a user input parameter obtained from the surrounding level ice.

3.4.2 Geometry Representation

Seidel (2016) developed an algorithm to store the element's geometry in the software applicable to all elements in the simulation, which are the rubble ice and the structure.

Their representation in three-dimensional space require information of the geometry, topology and direction of the normal vector of the faces. The geometry is the description of the boundaries of the element by defining the coordinates of the vertices of the polygon. The array *vert_coord_body* stores the coordinates of a given element. The topology is defined by two arrays; the first is the *face_vertex_table*, which contains the vertices that compose each face. The second array is the *vertex_face_table*, which contains the faces that each vertex belongs to.

To separate the exterior and interior of an element, the normal of a face should point outwards and three vertices of a face define the normal vector as described on the equation 21.

$$n = \frac{(v_2 - v_1) \times (v_3 - v_1)}{|(v_2 - v_1) \times (v_3 - v_1)|} \tag{21}$$

The rubble ice generation algorithm implemented is based on similar work developed by Metrikin (2014) who generates polygon shaped ice floes for the simulation of Stationkeeping of a vessel in ice floes.

The algorithm starts by determining the number of edges of the element from the beta random generator. After selecting the number of edges, the algorithm starts the first step of the rubble ice generation, which is the creation of a unit square centered at coordinates (0,0) as illustrated by Figure 24.

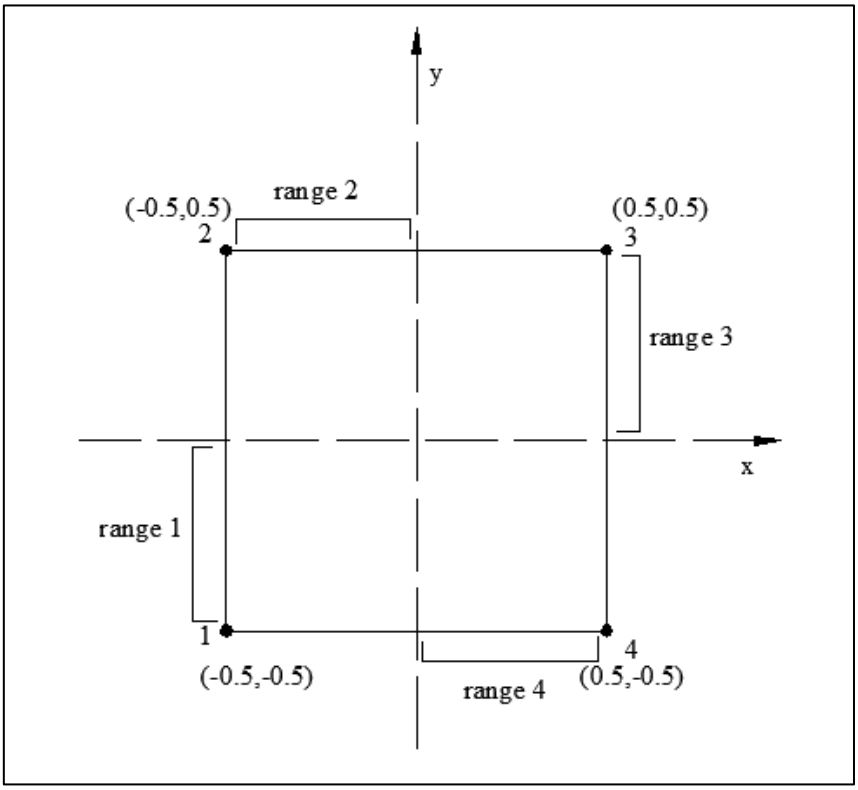

Figure 24. Initial unit square

The next step is to select the vertices randomly on each edge to create a quadrilateral from the initial square. The y coordinate of points 1 and 3 and the *x* coordinate of points 2 and 4 variate as shown in Figure 24. The quadrilateral becomes as shown in Figure 25. This procedure ensures that the minimum size of the particle is 50% of the axis-aligned bounding box area. The measurements of the particles from the images showed that, on average, the percentage of surface area in relation to the bounding box area is 65%.

It also guarantee that the points do not change quadrant so that the particle remains convex. As it is explained in section 2.1 the force calculation is obtained from the overlap between convex elements. According to Alekseev (2016) if a particle is concave, some of the contacts between particles may not be detected during the simulation.

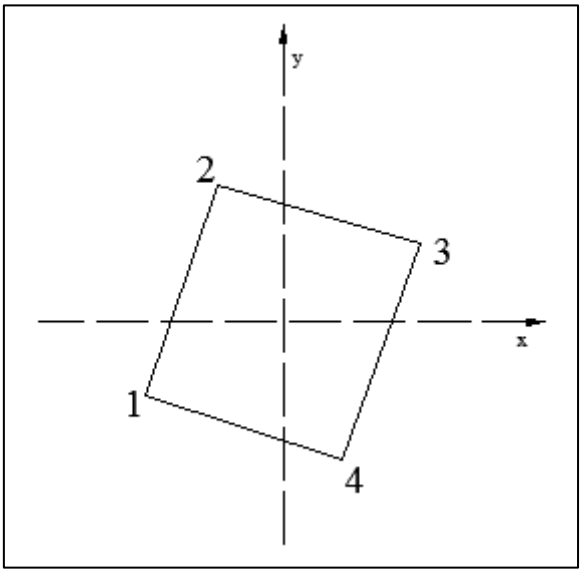

Figure 25. Rotated quadrilateral

If the number of edges of a given particle is 4, the algorithm goes to the scaling step; otherwise, more vertices are created according to number of edges within the bounding box.

Figure 26 shows a particle with five edges where the vertex number 2 is add. It has the same *x* coordinate as the previous vertex, in this example the vertex 1. And the *y* coordinate is randomly selected between [0 1] of the *y* coordinate of the following vertex, in this case vertex 3. The coordinates of the remaining vertices are maintained (vertices 1, 3, 4 and 5).

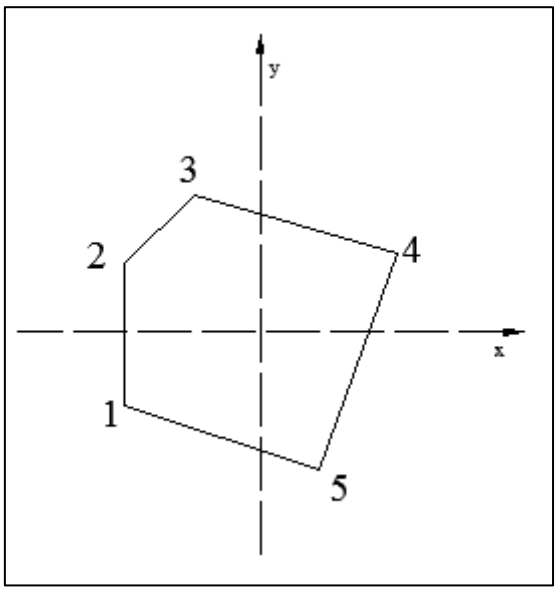

Figure 26. Five edges particle

In the case of the polygon with 6 edges as shown in Figure 27, the vertex 2 and 4 are created. The vertex 2 follows the same criteria explained before. The vertex 4 has the same *y* coordinate of the previous vertex (vertex 3). The *x* coordinate varies randomly from [0 1] of the *x* coordinate of the following vertex (vertex 5). The same concept is used to generate particles with 7 or 8 edges as illustrated in Figure 27.

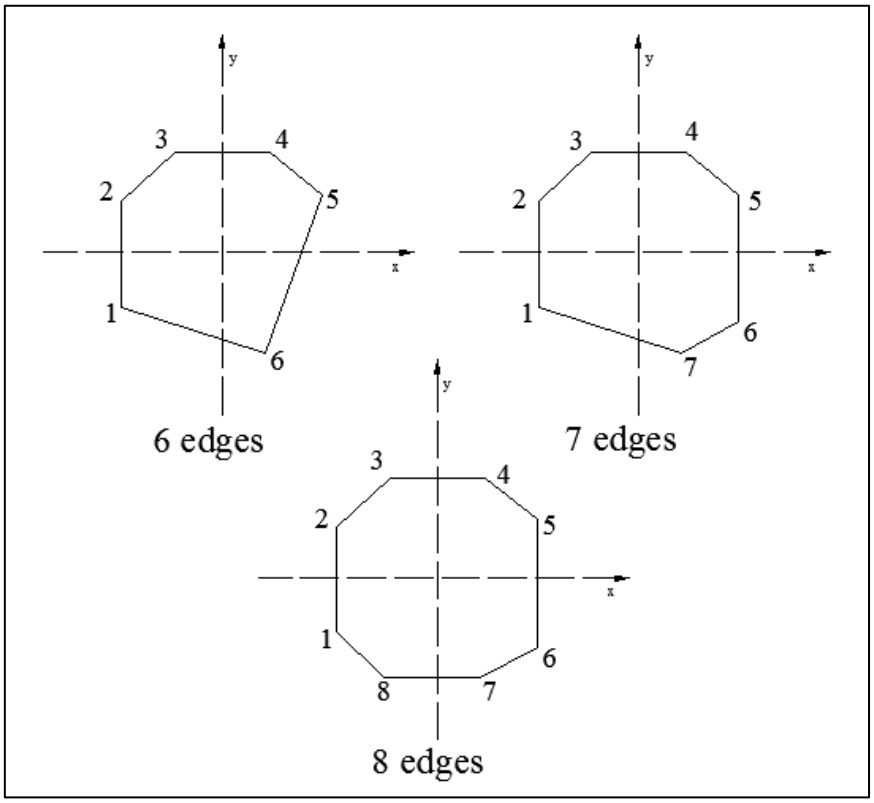

Figure 27. Rubble ice with 6, 7 and 8 edges

The following step is to scale and make a three-dimensional geometry array of the coordinates of the vertices. The coordinates in *x* and *y*-axis of all vertices are multiplied by the width and length, respectively, obtained from the beta random algorithm. This procedure ensures that the dimension of the axis-aligned bounding box is according to the probabilistic distribution curve. The *z* coordinate is symmetric to the plane *xy* based on user input value of thickness.

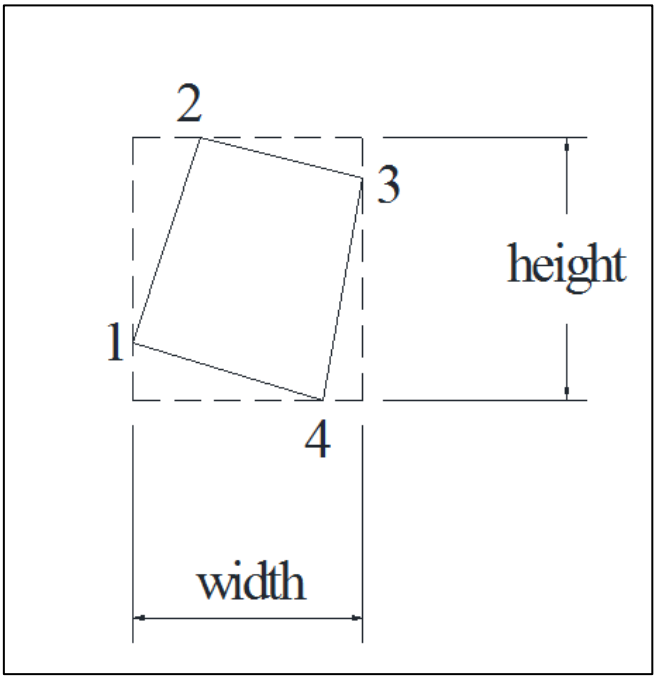

Figure 28. Scaling the coordinates of the rubble ice

Seidel (2016) distinguished two coordinate systems that exist in the code. The first is the Cartesian inertial space fixed system where all coordinates of all elements are introduced in the simulation. The second system in the co-moving located at the centre of mass of an element. The array containing the coordinates of the rubble ice is described in the co-moving system; therefore, the centre of gravity of the rubble ice has to be in the origin of this system.

After scaling the rubble ice, the centre of gravity may not be in the origin. The FORTRAN algorithm developed by Burkardt (2004) determines the centre of gravity of the particle as explained in section 3.5.2.

The coordinates of the vertices are subtracted from the coordinates of the calculated centre of gravity. Thus, the final centre of gravity of the polygon is moved to the origin. This procedure leads to the completion of the array *vert_coord_body* containing the three-dimensional coordinates of all vertices of the polygon in the co-moving coordinate's reference system.

3.4.3 Surface Mesh

The topology information contained in the arrays *face_vertex_table* and *vertex_face_table* should be a triangular mesh. The algorithm generate these arrays by identifying the faces and dividing it into triangles. The mesh is generated by taking the first vertex and create the remaining triangles from that vertex.

Figure 29 shows an example of meshed particle generated by the code for each number of edges possible.

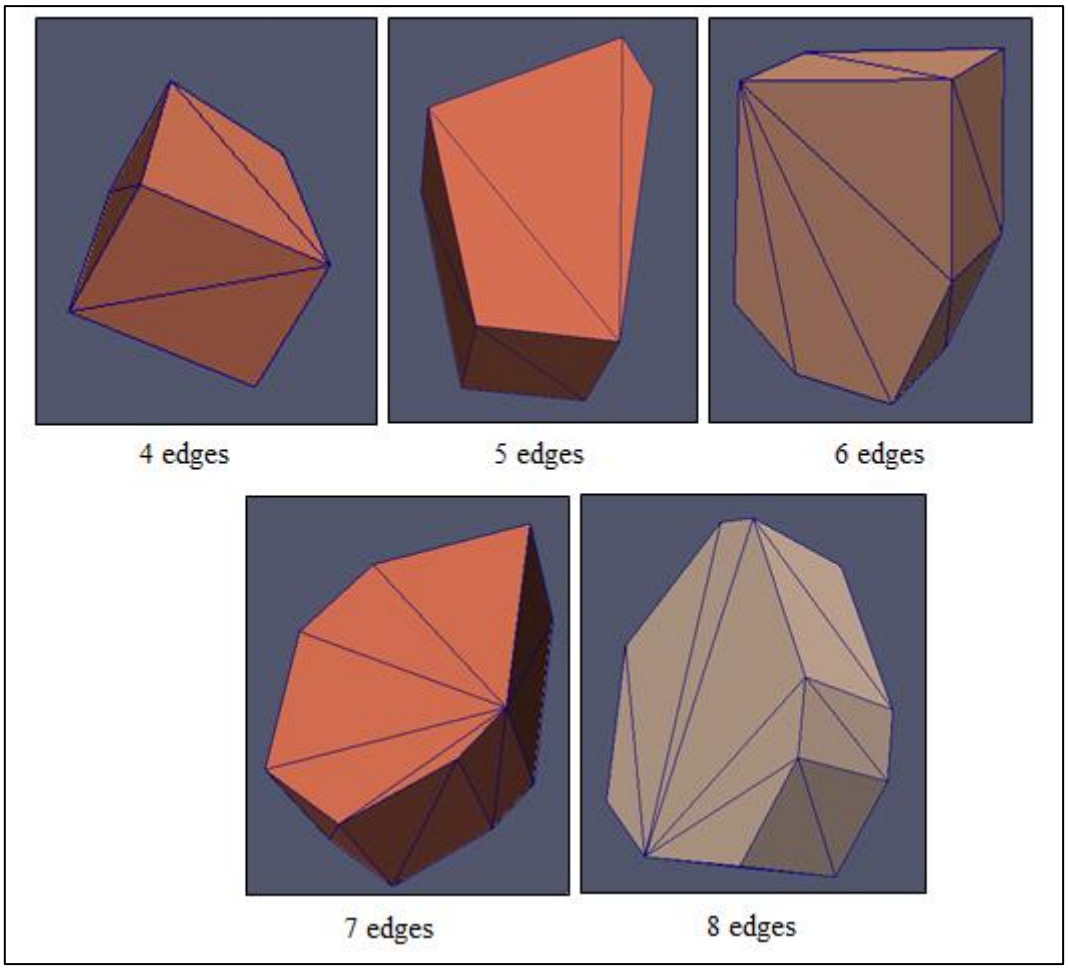

Figure 29. Meshed rubble ice

3.5 Rubble Ice Geometrical Properties

After defining the geometry of the rubble ice, the superficial area, volume, mass and moment of inertia of the element are calculated. The following sections discuss the algorithm implemented for the calculation of these parameters.

3.5.1 Rubble Ice Superficial Area

The superficial area of the polygon is the sum of the area of the triangles that compose each face by applying of the Schoelace formula that determines the area of a triangle based on the coordinates of the vertices. The FORTRAN algorithm is based on Burkardt (2005).

$$A_{tri} = \frac{1}{2}\left|\left(V_{1,x} - V_{3,x}\right)\left(V_{2,y} - V_{1,y}\right) - \left(V_{1,x} - V_{2,x}\right)\left(V_{3,y} - V_{1,y}\right)\right| \quad (22)$$

$$A_{face} = \sum_{1}^{n_{triangles}} A_{tri} \tag{23}$$

3.5.2 Rubble Ice Moment of Inertia, Volume, Mass and Centre of Gravity

The calculation of the moment of inertia, volume, mass and centre of gravity of a polygon is based on the coordinates of the mesh. The algorithm methodology is based on the work presented by Metrikin (2014).

The mesh of the polygon is expanded in three-dimensions by creating tetrahedrons where the first three vertices of the tetrahedron (t_1, t_2, t_3) are the vertices of the triangulated mesh of the face *i* and the fourth vertex is the origin of the co-moving coordinate system. For each tetrahedron generated, the coordinates are stored in the array *tetra.*

$$tetra^i = \begin{bmatrix} t_{1,1}^i & t_{2,2}^i & t_{3,1}^i \\ t_{1,2}^i & t_{2,2}^i & t_{3,2}^i \\ t_{1,3}^i & t_{2,3}^i & t_{3,3}^i \\ t_{1,4}^i & t_{2,4}^i & t_{3,4}^i \end{bmatrix} \tag{24}$$

The total volume of the polygon V_{total} is the sum of the volume of all tetrahedrons V^i

$$V^i = \frac{1}{6}\det(tetra^i) \tag{25}$$

$$V_{total} = \sum_i V^i \tag{26}$$

The mass of the rubble ice is calculated from the volume of the polygon and the density ρ_{ice} from the user input.

$$m_{ice} = V_{total} \cdot \rho_{ice} \tag{27}$$

The centre of gravity of a tetrahedron is calculated by the average of the coordinates

$$cg_{tetra} = \frac{1}{4}(\vec{t}_1^{\,\mathrm{i}} + \vec{t}_2^{\,\mathrm{i}} + \vec{t}_3^{\,\mathrm{i}} + \vec{t}_4^{\,\mathrm{i}}) \tag{28}$$

The centroid of the polygon is obtained from the centre of gravity of the tetrahedrons.

$$cg_{polygon} = \frac{1}{V_{total}} \cdot \sum_{i} cg_{tetra}.V^{i} \tag{29}$$

Tonon (2004) developed the equations of moment of inertia of a tetrahedron based on its vertices coordinates (t_1, t_2, t_3). An auxiliary three-dimensional vector $\overrightarrow{d^i}$ is used for the inertia calculation. The symbol $_o$ represents the Hadamard vector product.

$$\overrightarrow{d^i} = \left(\vec{t}_1^{\,i}\right)^2 + \left(\vec{t}_2^{\,i}\right)^2 + \left(\vec{t}_3^{\,i}\right)^2 + \vec{t}_1^{\,i} \circ \vec{t}_2^{\,i} + \vec{t}_1^{\,i} \circ \vec{t}_3^{\,i} + \vec{t}_2^{\,i} \circ \vec{t}_3^{\,i} \tag{30}$$

The moment of inertia of the tetrahedron is calculated as follow:

$$I_{1,1}^{i} = \frac{(d_y^i + d_z^i)}{10} \tag{31}$$

$$I_{2,2}^{i} = \frac{(d_x^i + d_z^i)}{10} \tag{32}$$

$$I_{3,3}^{i} = \frac{(d_x^i + d_y^i)}{10} \tag{33}$$

$$\begin{aligned} I_{1,2}^{i} = -\frac{1}{20}.\big(&2t_{1,1}^{i}t_{2,1}^{i} + t_{1,2}^{i}t_{2,1}^{i} + t_{1,3}^{i}t_{2,1}^{i} \\ &+ t_{1,1}^{i}t_{2,2}^{i} + 2t_{1,2}^{i}t_{2,2}^{i} + t_{1,3}^{i}t_{2,2}^{i} \\ &+ t_{1,1}^{i}t_{2,3}^{i} + t_{1,2}^{i}t_{2,3}^{i} + 2t_{1,3}^{i}t_{2,3}^{i}\big) \end{aligned} \tag{34}$$

$$I_{2,1}^{i} = I_{1,2}^{i} \tag{35}$$

$$\begin{aligned} I_{1,3}^{i} = -\frac{1}{20}.\big(&2t_{1,1}^{i}t_{3,1}^{i} + t_{1,2}^{i}t_{3,1}^{i} + t_{1,3}^{i}t_{3,1}^{i} \\ &+ t_{1,1}^{i}t_{3,2}^{i} + 2t_{1,2}^{i}t_{3,2}^{i} + t_{1,3}^{i}t_{3,2}^{i} \\ &+ t_{1,1}^{i}t_{3,3}^{i} + t_{1,2}^{i}t_{3,3}^{i} + 2t_{1,3}^{i}t_{3,3}^{i}\big) \end{aligned} \tag{36}$$

$$I_{3,1}^{i} = I_{1,3}^{i} \tag{37}$$

$$I_{2,3}^{i} = -\frac{1}{20}.\left(2t_{2,1}^{i}t_{3,1}^{i} + t_{2,2}^{i}t_{3,1}^{i} + t_{2,3}^{i}t_{3,1}^{i} + t_{2,1}^{i}t_{3,2}^{i} + 2t_{2,2}^{i}t_{3,2}^{i} + t_{2,3}^{i}t_{3,2}^{i} + t_{2,1}^{i}t_{3,3}^{i} + t_{2,2}^{i}t_{3,3}^{i} + 2t_{2,3}^{i}t_{3,3}^{i}\right) \quad (38)$$

$$I_{3,2}^{i} = I_{2,3}^{i} \quad (39)$$

Inertia tensor of the polygon about the origin is defined as:

$$I_0 = \sum_i I^i \cdot V^i \quad (40)$$

Then the inertia tensor is calculated referenced to the centre of mass using parallel axis theorem.

$$I_{tensor} = \rho_{ice}\left[I_0 - V_{total}\left(cg_{polygon}^2\, I_{unit} - cg_{polygon} \otimes cg_{final}\right)\right] \quad (41)$$

3.5.3 Algorithm Validation

To validate the algorithm that calculates the moment of inertia, volume and centre of gravity, five dummy particles with known geometry, one of each number of edges possible (between 4 and 8) were input in the software Rhino to calculate the same parameters.

The average difference from the results obtained for volume and centre of gravity is approximated to zero. The average of the results for the moment of inertia comparing to the FORTRAN code is shown on Table 6.

Table 6. Average difference Rhino and algorithm

Ixx	0.5842223%
Iyy	0.6242445%
Izz	0.0000057%
Ixy	0.0000030%

3.6 Rotational Dynamics

The inertia of the element is applied during the corrector step of the simulation for the calculation of the position, velocity, acceleration and quaternion of the element in the next time step. The angular momentum *L* is function of inertia *J*, and angular speed ω. Seidel (2016) detailed the calculation of the angular speed using quaternions.

$$\vec{L} = J\vec{\omega} \tag{42}$$

The first time derivative of angular speed $\dot{\omega}$ is dependent on the angular momentum *L*, angular velocity ω and torque τ.

$$\dot{\vec{\omega}} = J^{-1}(\vec{L} \times \vec{\omega} + \vec{\tau}) \tag{43}$$

The initial orientation of the particle is equal to the principal axes orientation. Therefore, its inertia tensor is diagonal during the initialization of the simulation. The elements are rotated in the co-moving system coordinate using the orientation quaternion at the previous time step.

4. NUMERICAL RIDGE SIMULATION

The software contains three options of simulation: ridge creation, punch test and ship simulation. The interface for the user is made through *txt*, *csv* and *obj* format files. The description of each simulation option is discussed in this chapter.

4.1 Ice Ridge Creation

The program creates an ice ridge according to the specifications of the ridge profile introduced by the user, which contains the geometry of the ridge and the mechanical properties of the rubble ice. The program outputs the complete ice ridge for later use in the simulation of a ship or a punch test. The ice ridge creation in the software does not follow the same principles as the natural formation as described on item 1.1 neither in the model basin described on item 1.2.

Figure 30 shows the ice ridge creation algorithm. The calculation of the geometry of the rubble ice described in chapter 3 happens in the initialization step, where the elements were placed in the global coordinates system by initializing the position, velocity and acceleration at the first time step of the ridge creation.

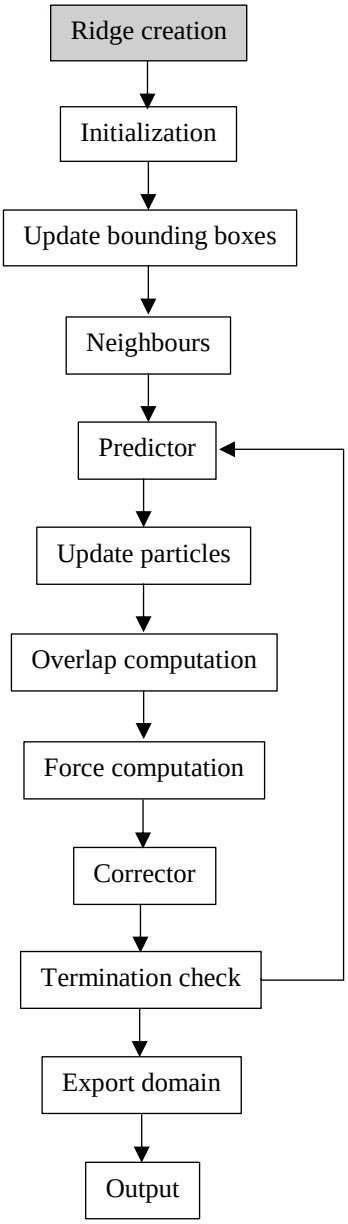

Figure 30. Ice ridge creation algorithm

The initial position of the centre of gravity of the particle is based on the maximum diagonal of all particles to avoid their overlap in the first step. The maximum diagonal d_{max} among all rubble ice elements is based on its axis-aligned bounding box main dimensions: length l_{ice}, width w_{ice} and thickness t_{ice}. Figure 31 illustrate this concept.

$$d_{max} = \max\left(\sqrt{l_{ice}^2 + w_{ice}^2 + t_{ice}^2}\right) \tag{44}$$

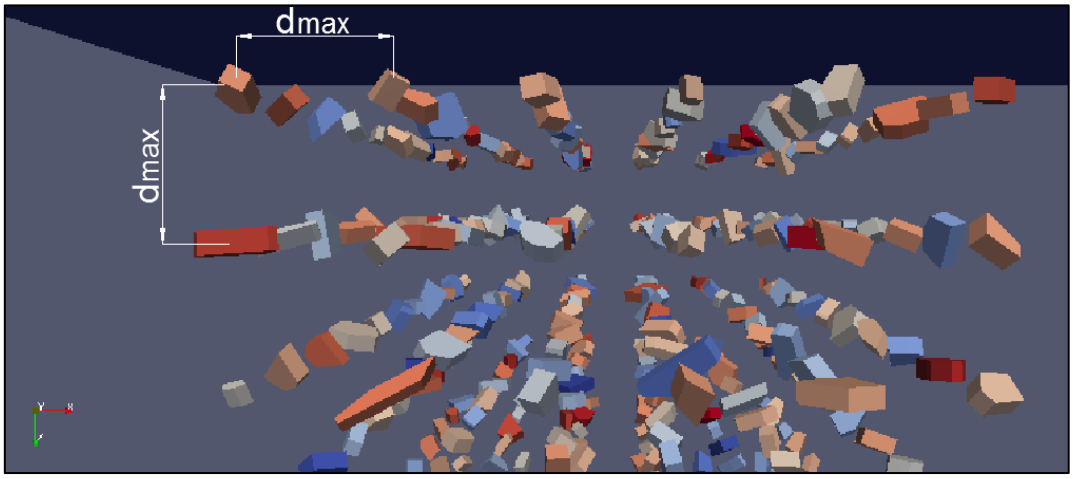

Figure 31. Initial distance between particles

The particles are placed in a row in *x* direction until the ridge width dimension is reached. Then a new row is created until the length of the ridge in *y* direction is completed. After the completion of one layer, a new layer on *z* direction is created. The loop is repeated until all the particles are placed as shown in Figure 32.

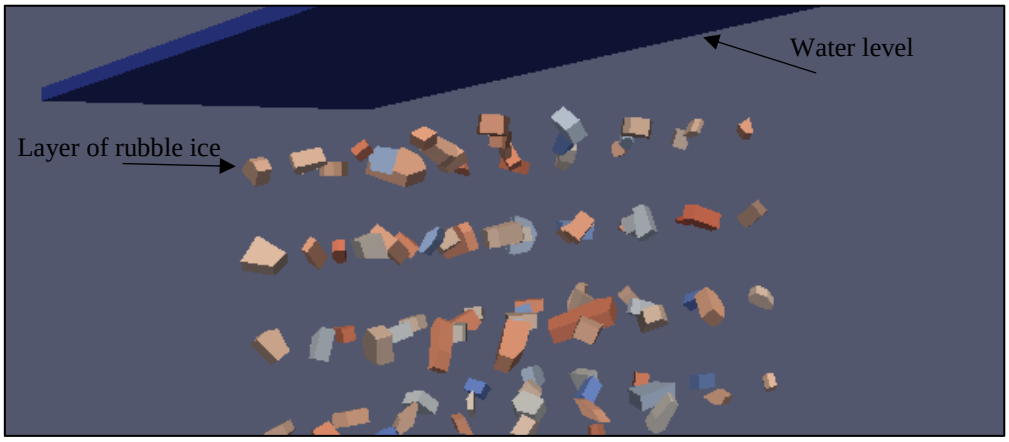

Figure 32. Initial position in ridge creation

After the initialization of the geometry of the particles, the software updates the axis-aligned bounding box to identify the pairs of particles in contact to calculate the forces. The software starts the main loop of the simulation with the predictor and update of particles. The force computation calls the algorithm for overlap computation, which provides all the information discussed in item 2.2 for force calculation.

During the main loop, the rubble ice float up due to its buoyancy until all particles are in contact with the consolidated layer. Two auxiliary planes are created in order to compact the geometry of the ridge according to the input ridge profile as shown in Figure 33.

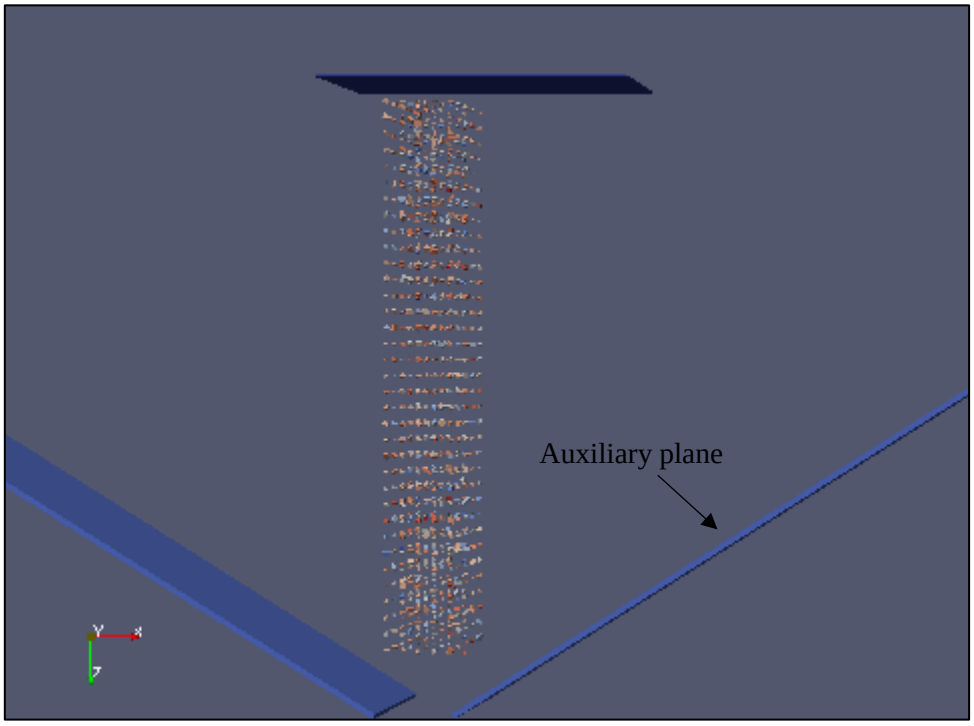

Figure 33. Auxiliary planes for ridge creation

The algorithm exports position, quaternions, moment of inertia, geometry, velocities, acceleration and physical properties of the rubble ice in *txt* format files. When a punch or a ship is simulated, this information is imported before the main calculation loop so there is no need to create a new ridge for every structure simulation, allowing the usage of a given simulated ice ridge several times with different structures.

The ridge simulation ends when the termination check identifies that the auxiliary planes had reversed during 2 seconds. The condition to reverse the planes is that the distance at the water level between the planes is the equal to the ridge width determined by the user as shown in Figure 34. The planes reverse to allow proper arrangement of the rubble ice in the ridge keel as it is shown on the final ridge geometry in Figure 35.

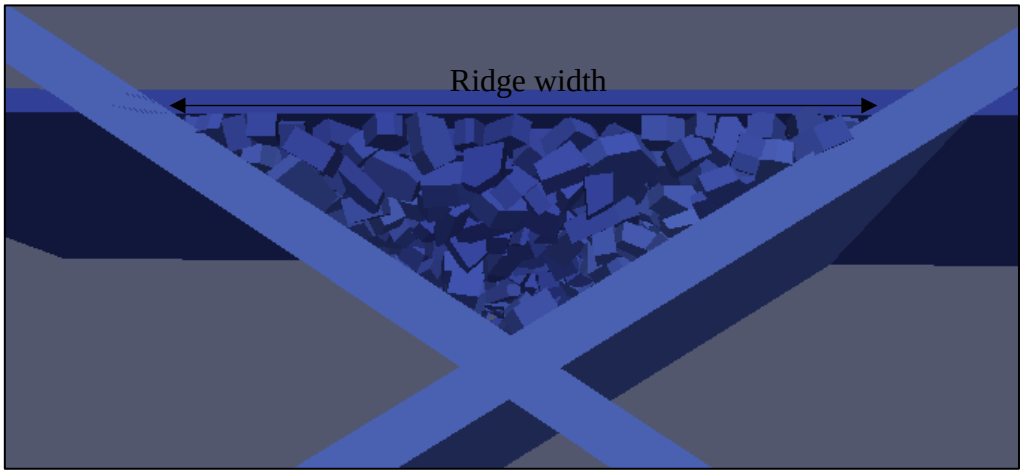

Figure 34. User input ridge profile

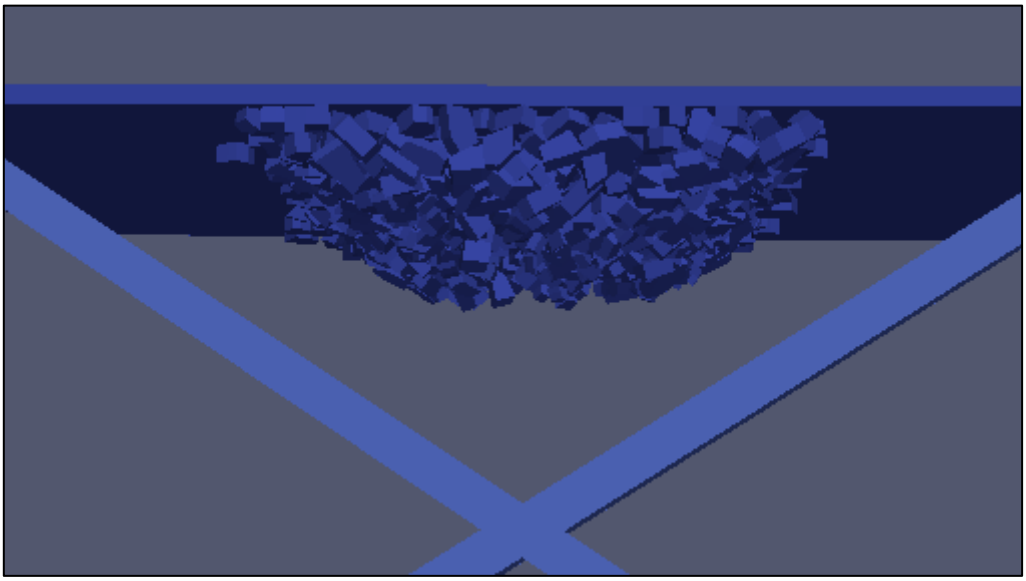

Figure 35. Final ridge profile

4.2 Punch Test Simulation

To run a punch test simulation, the software imports the ice ridge properties before the main loop calculation starts. The calculation steps are similar to the ridge creation as described in Figure 36, the main calculation loop consists of predictor, update, overlap computation, force computation, corrector and termination check. The simulation outputs the forces, acceleration, position and velocity of the punch device.

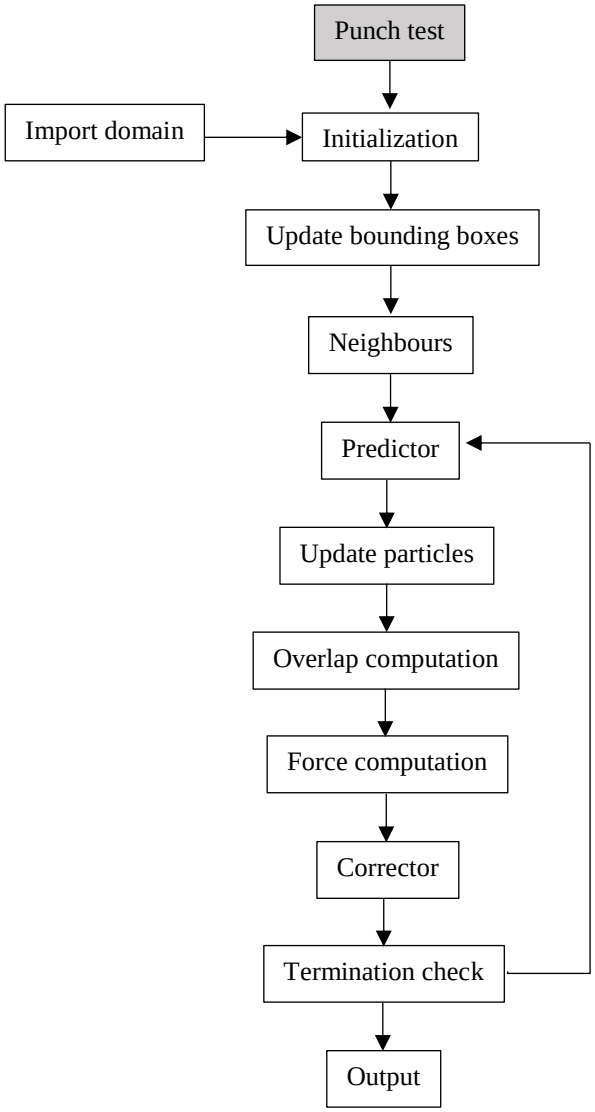

Figure 36. Punch test algorithm

4.2.1 Punch Test Simulation Input

The structure representing the punch device is input in the simulation as an *obj* format file containing the vertices and faces of the object. For contact calculation, the surfaces of the structure have to be made of triangular mesh following the criteria explained by Alekseev (2016). Figure 37 shows an example of the mesh of the punch device in the simulation.

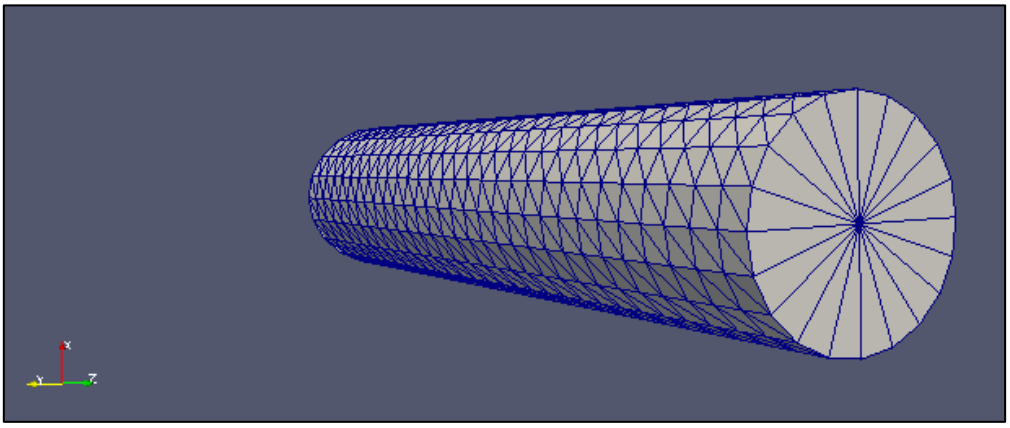

Figure 37. Punch device mesh

To assign values to the width, length, height and mass of the structure, a *txt* format file containing the information is read by the code. Since the device is a cylinder, the length and width of the structure refers to its diameter. The height refers to the dimension in the *z* direction. The mass of the punch device is the same as used in the experimental testing.

The structure is positioned above the level ice in the first time step and the constant downward speed in the *z* direction is 7 mm/s. The vertical speed simulated is the same used during the experiment. The Young's modulus of the punch device is the same as structural steel ASTM-A36 of 200 GPa. The summary of the punch device simulated in the simulation is described in Table 7

Table 7. Punch device properties summary

Properties / Punch diameter	Length	Width	Thickness	Mass	Young's modulus
0.18 m	0.18 m	0.18 m	1.0 m	88.5 kg	200 GPa
0.25 m	0.25 m	0.25 m	1.0 m	122.0 kg	200 GPa

The ridge profile information was obtained from the ice tank test performed at HSVA, which includes the dimension of ridge width, keel width, keel height and thickness of the rubble ice. The length of the ridge simulated was 2.0 m. The simulated ice ridge should be long enough to avoid interference from the boundaries; however, the length should not be too long so that the simulation is done within a reasonable time. In the example analysed, the rubble ice thickness was 20 mm. The final ridge profile in the simulation is shown in Figure 38.

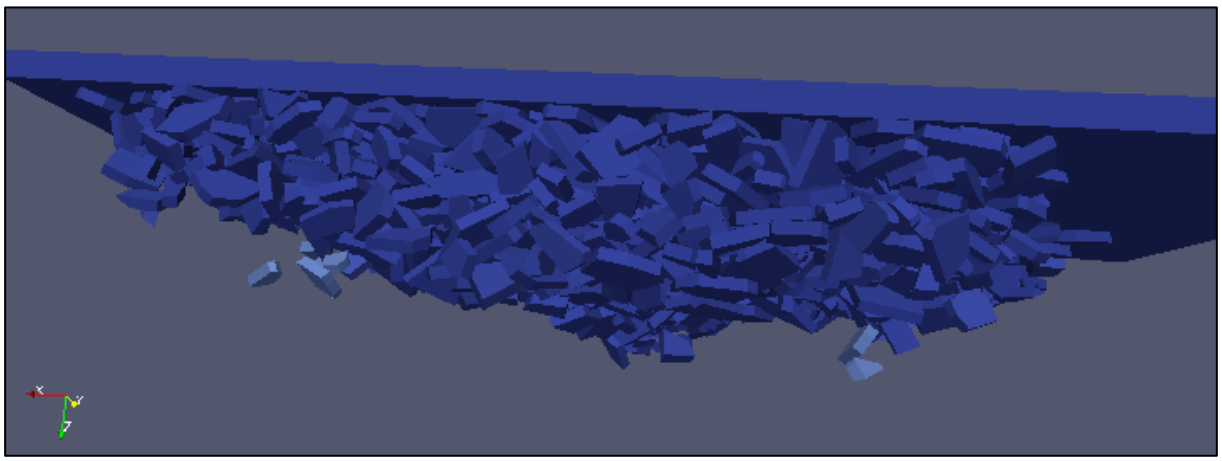

Figure 38. Simulated ice ridge

Table 8. Input parameters of the ice ridge for punch test simulation

Ridge width	1.50 m
Ridge keel width	0.313 m
Ridge keel height	0.375 m
Ridge length	2.00 m
Void fracture	55%
Rubble thickness	0.02 m
Ice density	900 kg/m³
Ice Young's modulus	90900 Pa
Ice bending strength	50000 Pa
Ice Poisson's ratio	0.30
Cohesion coefficient	0.0001
Viscous damping coefficient	2.0
Normal damping force coefficient	0.2
Tangential dissipation force coefficient	0.2
Ice friction coefficient	1.0
frac	0.1

The ice density, Young's modulus, bending strength and Poisson's ratio of the ice were obtained from HSVA ice tank testing. Alekseev (2016) calibrated the forces coefficients for cohesion, viscous damping, normal damping, tangential dissipation and friction for rectangular shape rubble ice simulation and the values were kept the same for this simulation with polygon shape.

The last parameter is the *frac* coefficient for DEM time step calculation. Hart et al. (1988) calibrated *frac,* which represents that an element may be in contact with several elements simultaneously.

The simulation ends when the base of the punch device penetrates 20 cm below the keel height or the pre-determined total number of steps of the simulation is completed.

4.2.2 Punch Test Simulation Results

To validate the simulation, data from experimental punch test performed at HSVA is compared with the numerical simulation results. The force measured in the vertical axis F_{ridge} is the sum of the weight of the cylinder w_c, the buoyancy of the cylinder b_c, the weight of the rubble ice underneath the cylinder w_i and the buoyancy of the rubble ice b_i. The sum of the forces from the interaction between cylinder's wall and rubble ice is defined as shear force s_i.

$$F_{ridge} = (w_c - b_c) - (w_i - b_i) - s_i \tag{45}$$

In order to remove the weight and buoyancy of the cylinder in the data analysis, the cylinder is lowered in open water with the same vertical velocity as in the ridge punch test. Then, the open water test results are subtracted from the forces measured in the ridge punch tests. This procedure is performed in the experimental test in the ice basin and in the simulation. Figure 39 illustrate the forces for punch test and open water measurements.

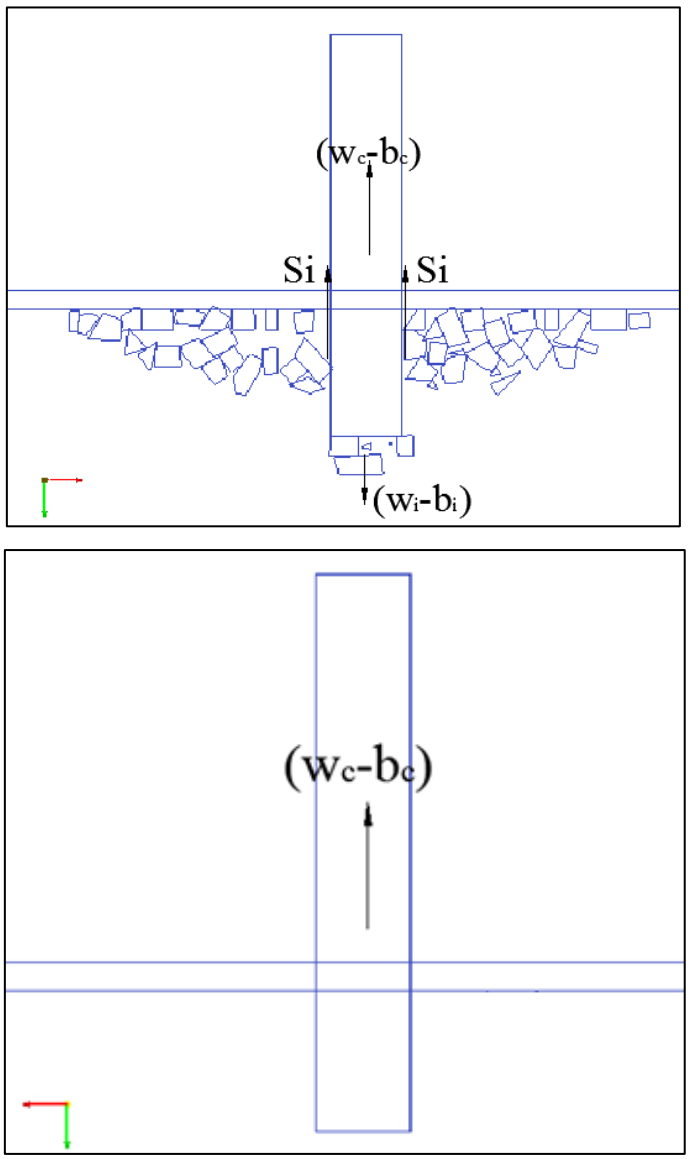

Figure 39. Punch test forces in ice ridge (top) and open water (bottom)

To explain further details regarding the results of the simulated punch test, Figure 40 shows two phases of a punch test in ice tank. In Phase I, the punch is lowered down and gets in contact with the rubble ice causing a peak of forces due to the freezing bonds between them.

According to Serré (2011), those bonds are created due to the buoyancy that keeps the rubble ice in close contact. When the ice blocks comes to rest after the ridge formation, the water trapped in the contact points freezes.

During phase II, there are no freezing bonds and the edges of the rubble ice are worn due to the friction, therefore the force decrease with the immersion of the rubble ice under the punch. The punch continues to push the rubble ice out of the ridge, consequently the amount of ice particles that are in contact reduces every time step causing less friction forces between ice-ice. Figure 41 illustrate the punch test in ice basin.

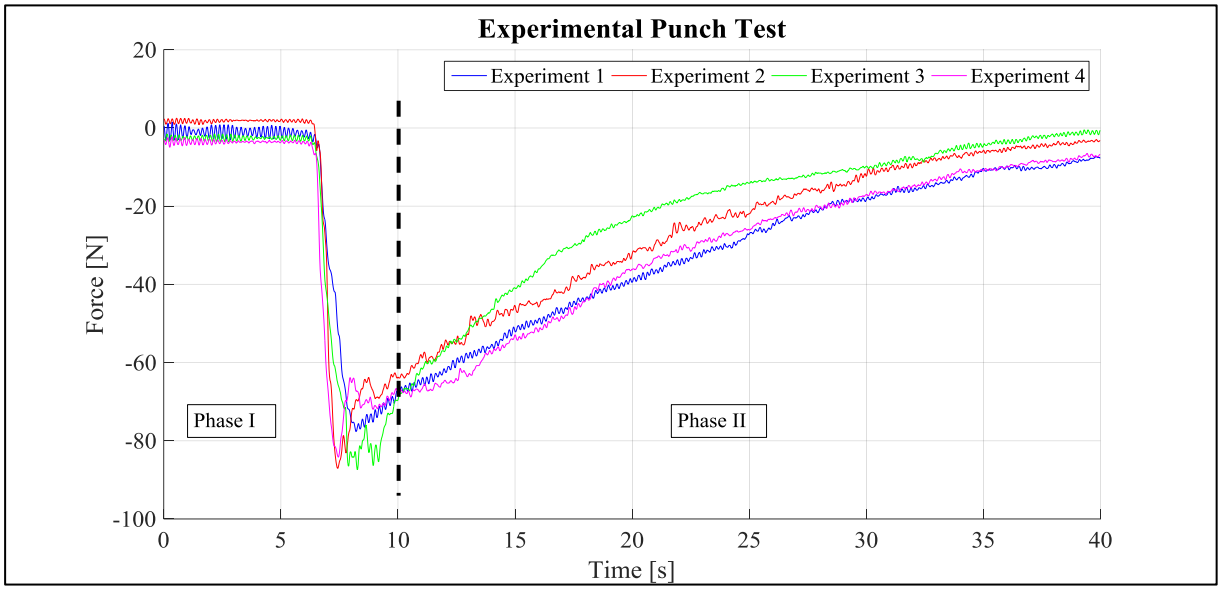

Figure 40. Experimental punch test

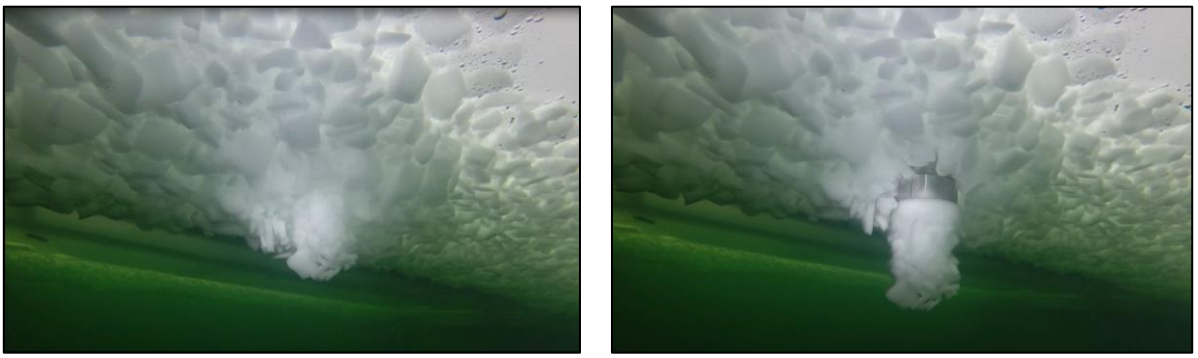

Figure 41. Punch test in ice basin

Figure 42 shows two numerical simulations and the average of the experimental results of Figure 40. One simulation was run with rectangular particle generation and the other simulation has the implemented polygon shape geometry to represent the rubble ice. The punch device has a diameter of 180 mm.

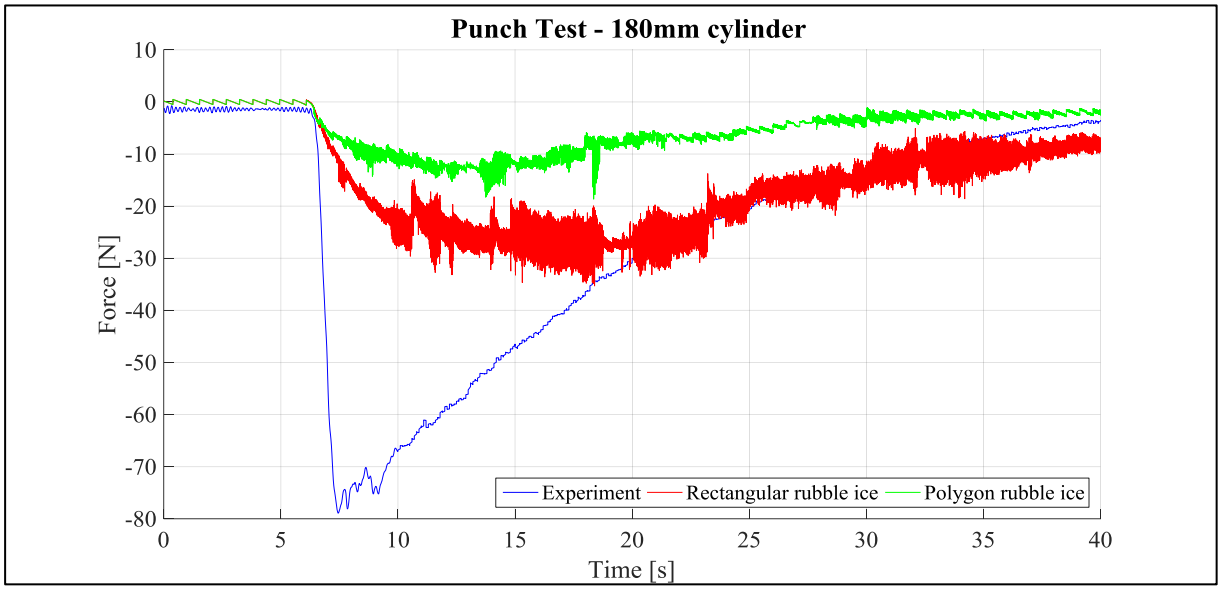

Figure 42. Simulation and experimental punch test for 180 mm cylinder

The same ridge geometry was simulated using the cylinder of 250 mm diameter. The result is shown in Figure 43.

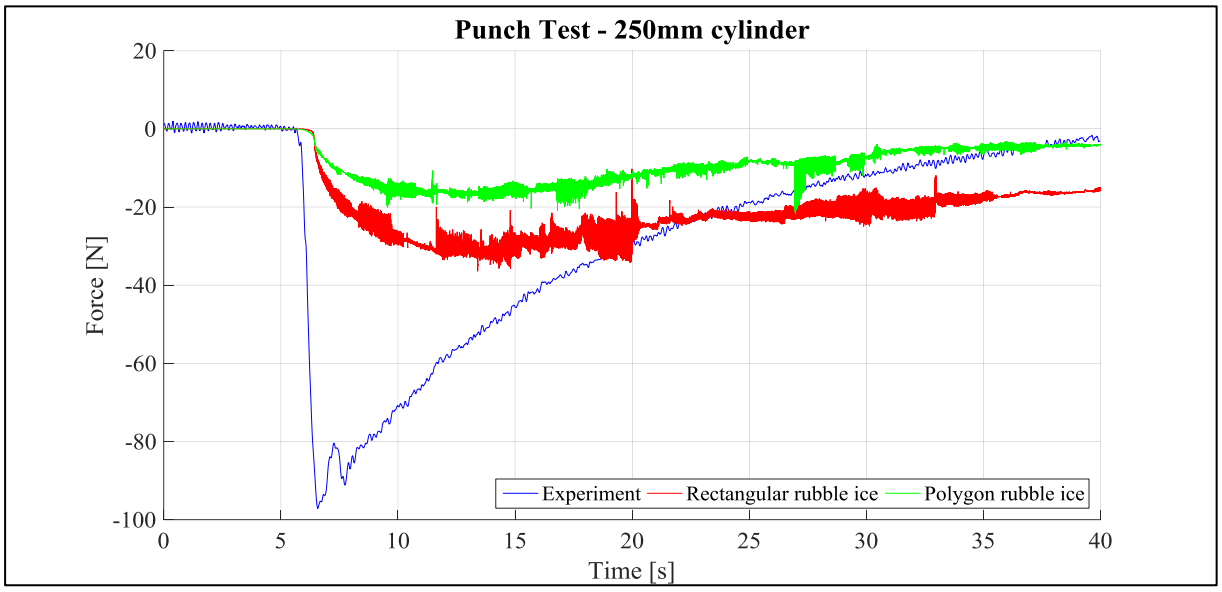

Figure 43. Simulation and experimental punch test for 250 mm cylinder

The results obtained from the simulation with the polygon shape geometry shows that the measured force opposing the cylinder's movement is smaller than the simulation with rectangular shape rubble ice and the experiment. One of the reasons is the increase in the number of rubble ice pieces underneath the punch device as shown in Figure 44, which creates buoyancy forces on the punch. Furthermore, the input force coefficients described on Table 8

are the same as the rectangular shaped geometry. Calibrating the force coefficient could approximate both simulation curves.

The comparison between simulation and experiment showed that the numerical simulation does not reproduce the phase I of the punch test. The aforementioned freezing bonds are not modelled in the force calculation in the software in any of the versions of the code; for this reason the peak of force measured in the ice tank test is not recreated in the simulation.

The main improvement of generating polygon shaped particles in the simulation is the qualitative behaviour of the rubble ice particles during the punch test. In the simulation with polygon shape rubble ice, the amount of ice particles underneath the punch device increased due to the large variability of rubble ice size and shape. Another improvement is that the implementation showed smoother force measurements than before.

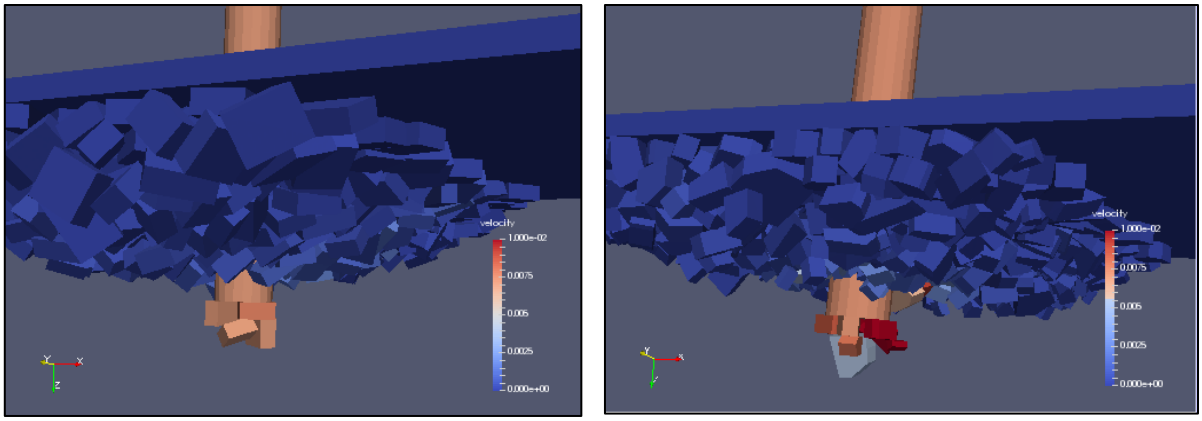

Figure 44. Punch test simulation rectangular shape (left) polygon shape (right)

To compare the rubble ice dimensions in between both simulations, Figure 45 shows the bounding box height for polygon and rectangular rubble ice shape. The implemented feature allows the coverage of the range of sizes of the rubble ice that the experimental ice tank would generate.

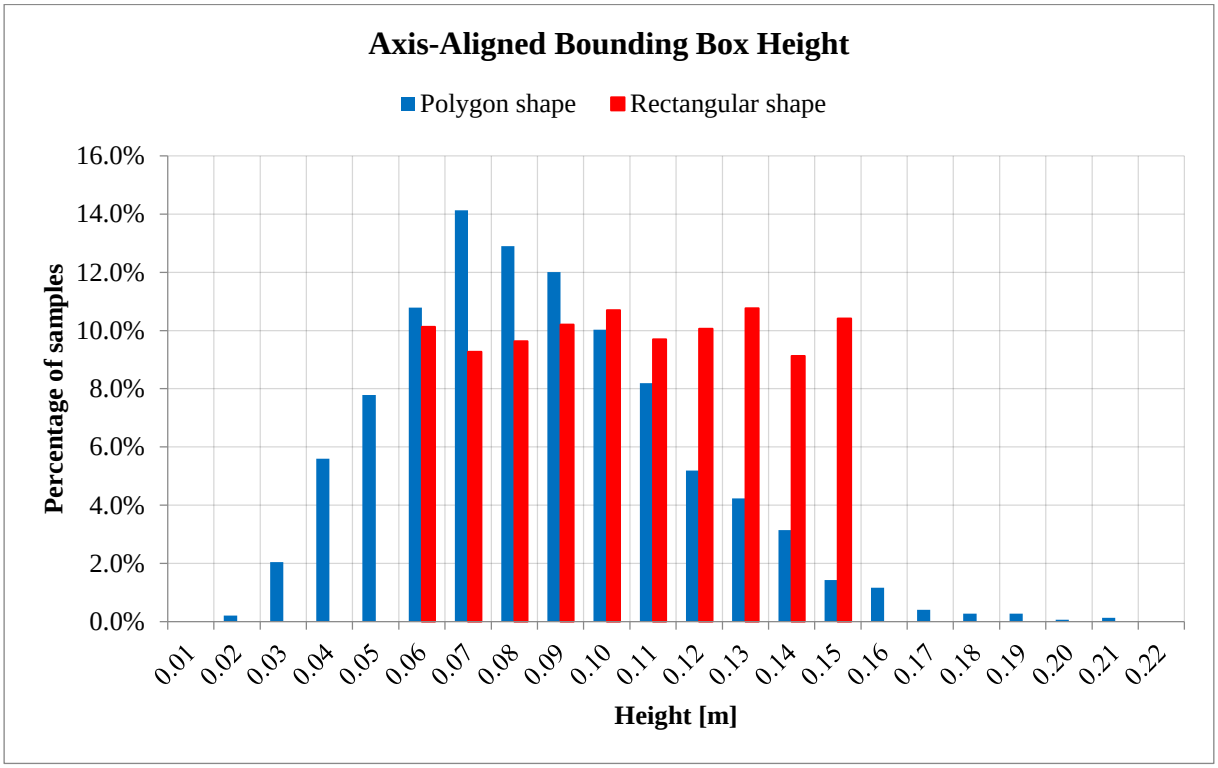

Figure 45. Rubble ice height distribution

In order to investigate the influence of the rubble ice thickness in the results of the punch test, two simulations were performed with both versions of the code. The ridge profile and mechanical properties of the ice were kept the same. The only modification was the increase of the rubble ice thickness to 50 mm.

Figure 46 shows the result from the simulation of punch test with 50 mm and 20 mm ice thickness in the software version that generates rectangular shapes to represent the rubble ice, and the Figure 47 shows the results for the same comparison for polygon shaped rubble ice.

The simulation with 20 mm rubble ice thickness showed higher forces than 50 mm thickness because the number of rubble ice elements increased to maintain the same ridge volume. Therefore, there is an increase in the number of contacts between rubble ice and cylinder during the simulation. This fact is more evident for rectangular shape simulation.

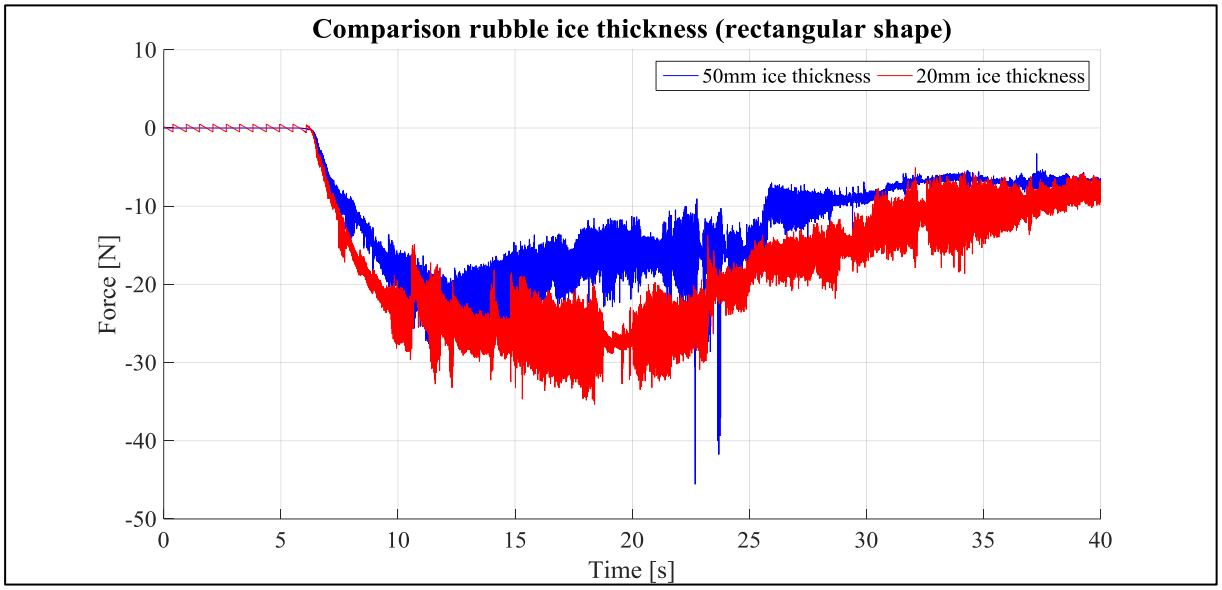

Figure 46. Thickness of the rubble ice comparison with rectangular shape

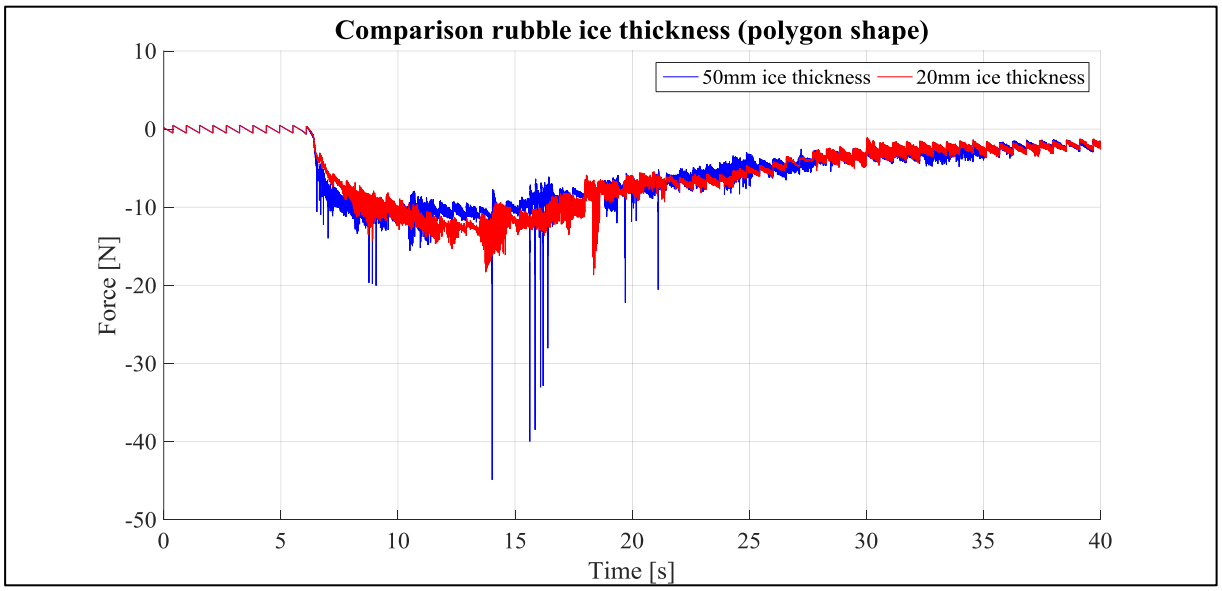

Figure 47. Thickness of the rubble ice comparison with polygon shape

The main drawback of the new implementation is that the computational time with the polygon shaped geometry increased significantly due to the meshed faces of the particles causing larger amount of overlaps to be calculated.

4.3 Ship Simulation

Alekseev (2016) introduced in the code the algorithm to simulate a ship interacting with an ice ridge. The initialization of the program is similar to the punch test simulation, where the

algorithm imports the ridge profile and the structure of the ship before starting the main loop calculation.

As shown in Figure 48, the ship buoyancy is calculated before the main loop to save computational time. The algorithm sorts different combinations of draft, pitch, roll angle and store the calculated buoyancy values. During the simulation, the values of the mentioned parameters are interpolated to obtain the current buoyancy of the ship. The same procedure is performed for the calculation of buoyancy restoring moment.

At the first time step in the ship simulation, the ship's centre line is coaxial to the centre of the ridge and the bow is a few centimeters from the ridge. The initial horizontal velocity is obtained from the measured arrival velocity at the ridge during the experimental ice tank test; so that the ship in the simulation has the same initial kinetic energy as in the experiment.

The ship simulation ends when the minimum x coordinate of the ship is 50 cm from the ridge or the predefined maximum number of steps of the simulation is completed.

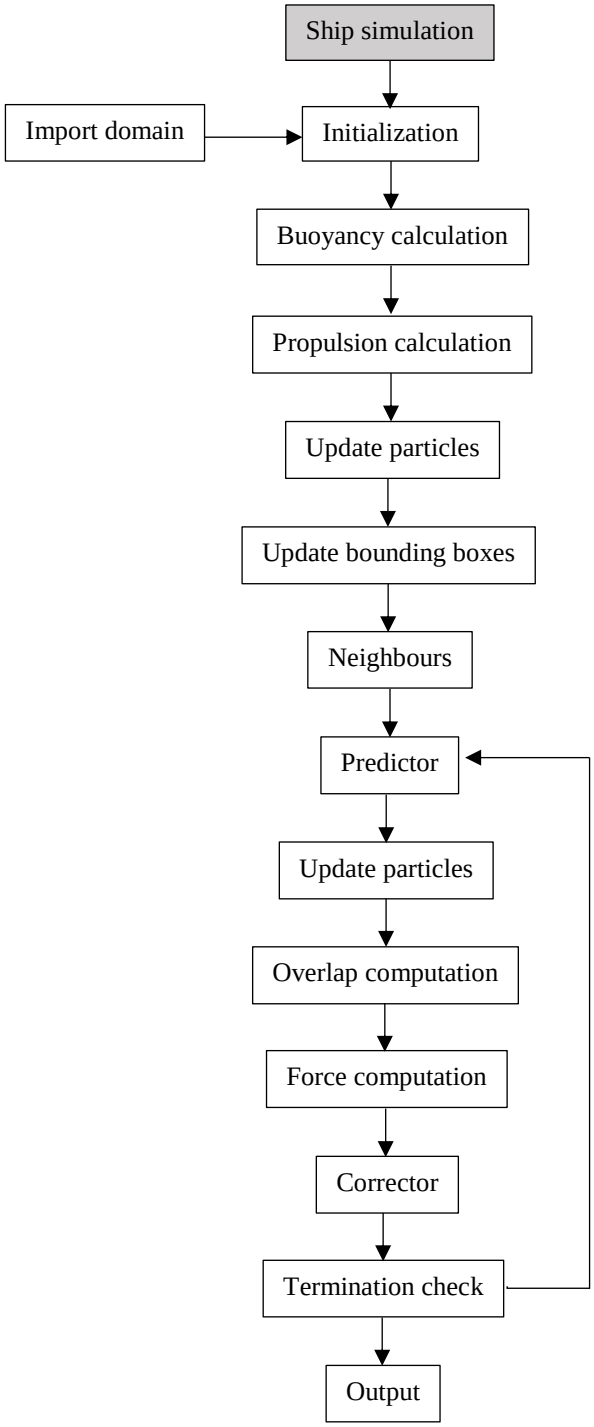

Figure 48. Ship simulation algorithm

4.3.1 Ship Simulation Input

The ship model is a non-convex geometry requiring the division into convex subparts as shown in Figure 49. The appendages of the ship, such as the rudder and shaft brackets are removed from the model to simplify the geometry. The subparts are positioned in the global coordinate system based on its centre of gravity informed through a *txt* file.

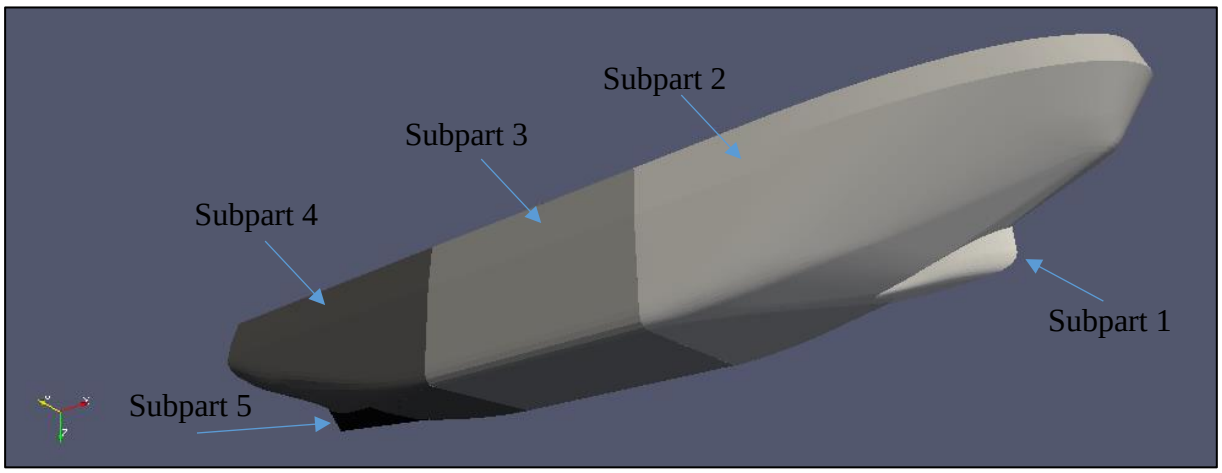

Figure 49. Ship structure

Each subpart of the ship model is imported through *obj* format file containing the vertices and faces. As described for the punch model, the geometry of the ship is composed of triangular mesh without discontinuities.

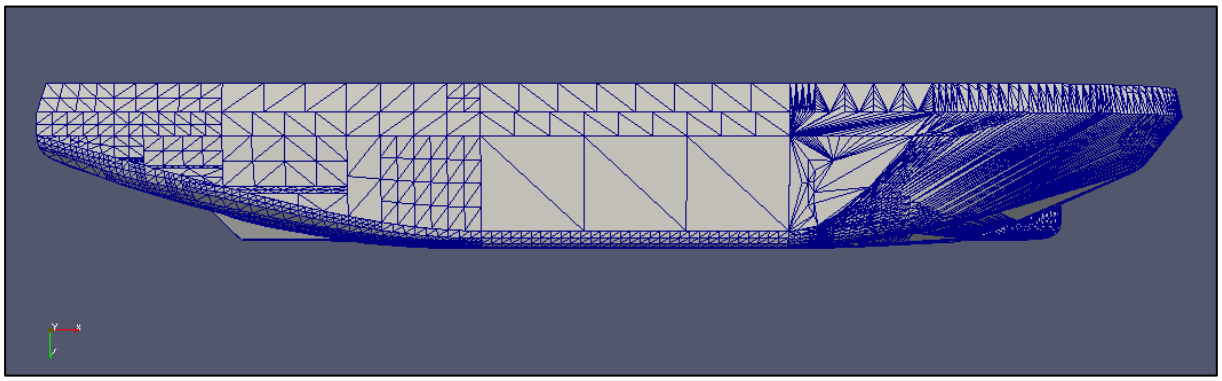

Figure 50. Ship mesh

Another *txt* format file contains all the remaining information necessary for the simulation, which are described in Table 9. The values of the parameters described in the table are for the model scaled ship that was tested in the ice tank; therefore, the same dimensions of the ship's model are simulated in the software. The scale factor for this model is $\lambda = 22$.

Table 9. Ship input properties for simulation

Number of subparts	5
KG	0.461 m
Waterline length	5.754 m
Beam	1.045 m
Draft	0.909 m
Mass	1433.13 kg
Ice resistance curve (Lindqvist model)	R_{ice} = 264.32 (V_{ship}) + 140.60
Friction coefficient of the ship's hull	0.2
Young's modulus of the ship hull;	$200 \cdot 10^9$ Pa
Initial position of CG	[-3.013, 1.569, -0.120] m
Initial velocity	[0.37, 0, 0] m/s

The thrust calculation is based on the propeller curve obtained from open water test. The user inputs the parameters to define the geometry of the propeller through *txt* format file containing the number of propellers, diameter, rotational speed and wake fraction coefficient. In addition, the user inputs a *csv* file containing the non-dimensional parameters: advance ratio, thrust coefficient and torque coefficient. Those parameters allow the calculation of the propeller thrust during the simulation based on the advance velocity of the ship at each time step. The propeller characteristics used in the simulation is described in Figure 51.

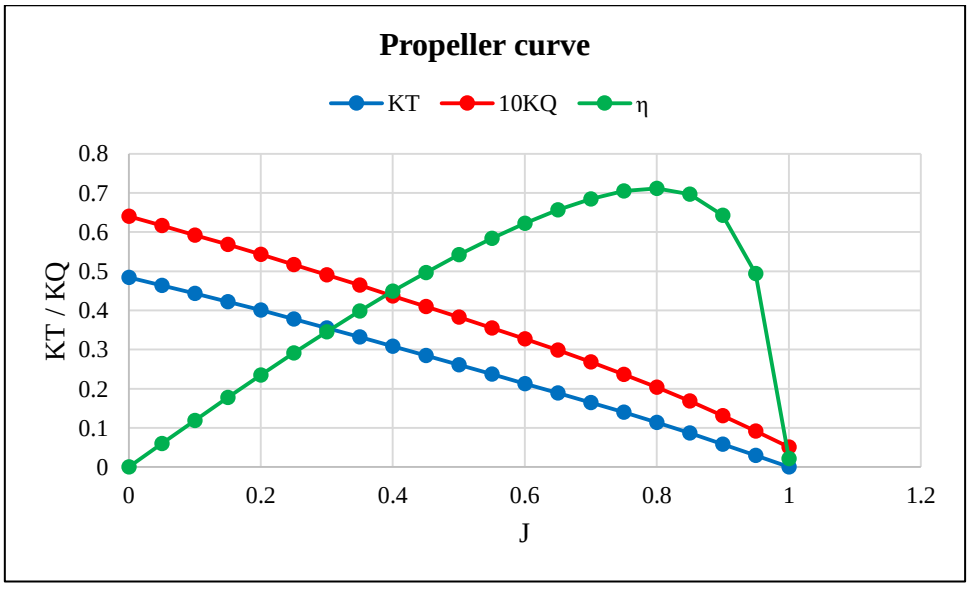

Figure 51. Propeller curve of the simulated ship

The ice ridge profile in the simulation analysed is the same for the ice ridge constructed for the ice tank test. The ice ridge length in the simulation is three times longer than the width of the ship to avoid any influence from the borders of the ice ridge. The parameters of the ship simulation are summarized on Table 10. Note that the ice ridge keel width is zero, which indicates that the ice ridge profile has triangular shape. The scaling factor of the ice ridge is the same as the ship model, $\lambda = 22$.

Table 10. Input parameters of the ice ridge for ship simulation

Ridge width	1.364 m
Ridge keel width	0.000 m
Ridge keel height	0.314 m
Ridge length	3.137 m
Void fracture	55%
Rubble thickness	0.0415 m
Ice density	900 kg/m³
Ice Young's modulus	90900 Pa
Ice bending strength	50000 Pa
Ice Poisson's ratio	0.30
Cohesion coefficient	0.0001
Viscous damping coefficient	2.0
Normal damping force coefficient	0.2
Tangential dissipation force coefficient	0.2
Ice friction coefficient	1.0
frac	0.1

4.3.2 Ship Simulation Results

Figure 52 shows the measurements of the ship's velocity from the simulation using the rectangular shaped rubble ice and the implemented polygon shape. The results from the ice tank experiment is used as a reference for both simulations.

The ship arrives at the ice ridge with 0.37 m/s of velocity. The velocity of the ship drops to a minimum value and after the bow of the ship penetrates the ice ridge, the ship is able to increase its speed and overcome the ice ridge at once.

At the simulation time 13.5 seconds, the velocity reaches the lowest measured value, 0.11 m/s. The ship's position at this speed is showed in Figure 53, where the start of the parallel body section is at the middle of the ridge. This result agrees with Ehle (2012) research on ship

resistance in model ice tank. The author concluded that the highest resistance value is obtained when the ship's fore shoulder is at the maximum keel depth. As the bow leaves the ridge, the ship is able to increase the velocity again.

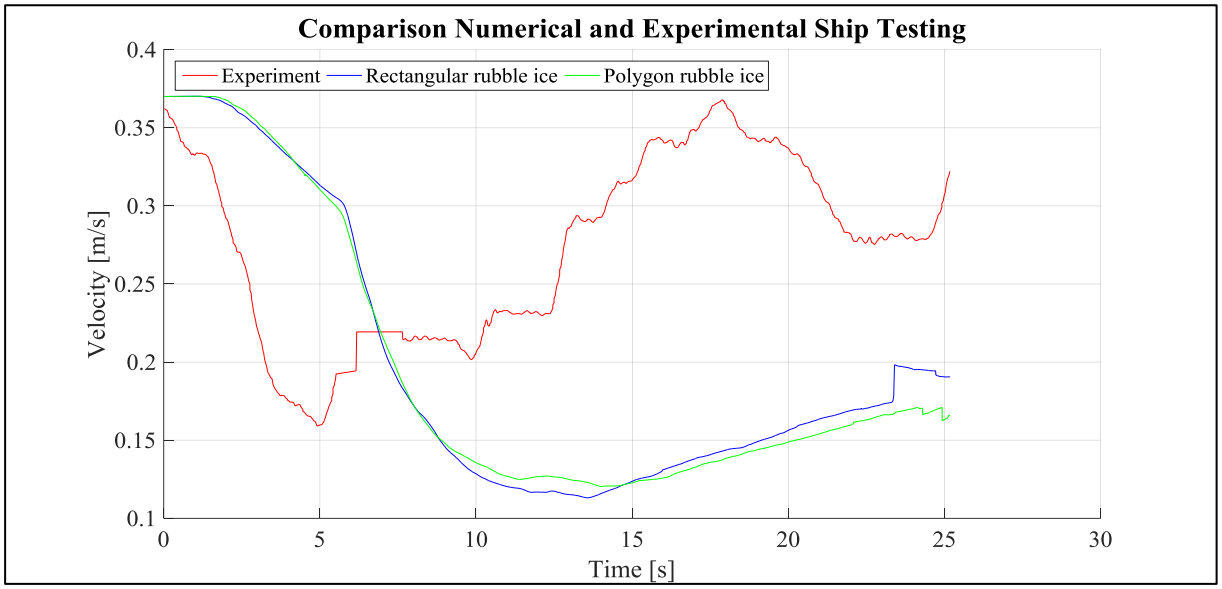

Figure 52. Ship simulation results

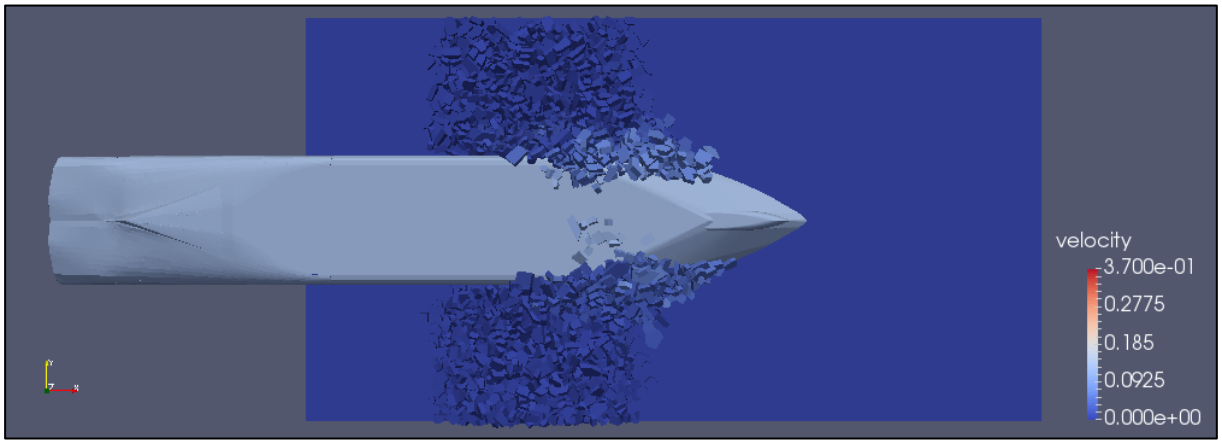

Figure 53. Bottom view, position at the lowest velocity measured

The implemented feature in the software did not influence the ship velocity quantitative results as shown in Figure 52. Thus, the conclusion is that the rubble ice particle shape does not influence the ship and ice ridge interaction.

Conversely from the punch test, the freezing bonds does not have significant influence in the ship simulation even though the freezing bonds are not modelled. The reason is that the velocity of the ship is large compared to the punch device so the force necessary to break the

freezing bonds is small compare to the all remaining forces mentioned in section 2.3 that takes place when a ship interacts to the ice ridge.

The images from Figure 54 to Figure 59 compared the qualitative results of the simulation with rectangular and polygon shape at the simulation time 25 seconds. The bottom view images shows the rubble ice attached to the hull is more accurate with the implemented feature. The designer of the hull is interested on the flow pattern of the rubble ice flow towards the stern.

In addition, the agreement of the velocity pattern between simulation and ice tank test indicates that the friction model used is accurate.

For instance, the simulation does not take into account the propeller-ice interaction nor the appendages, such as the rudder. Those components of the ship influence on the ice ridge resistance in two case: when the stern of the ship approaches the ice ridge or when a ship breaks the ice ridge astern.

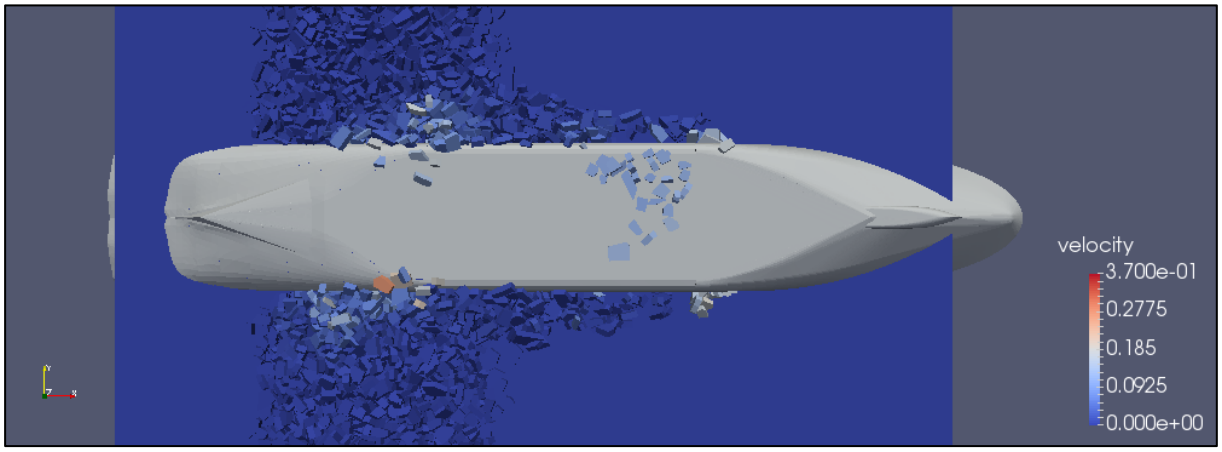

Figure 54. Bottom view, polygon shaped rubble ice

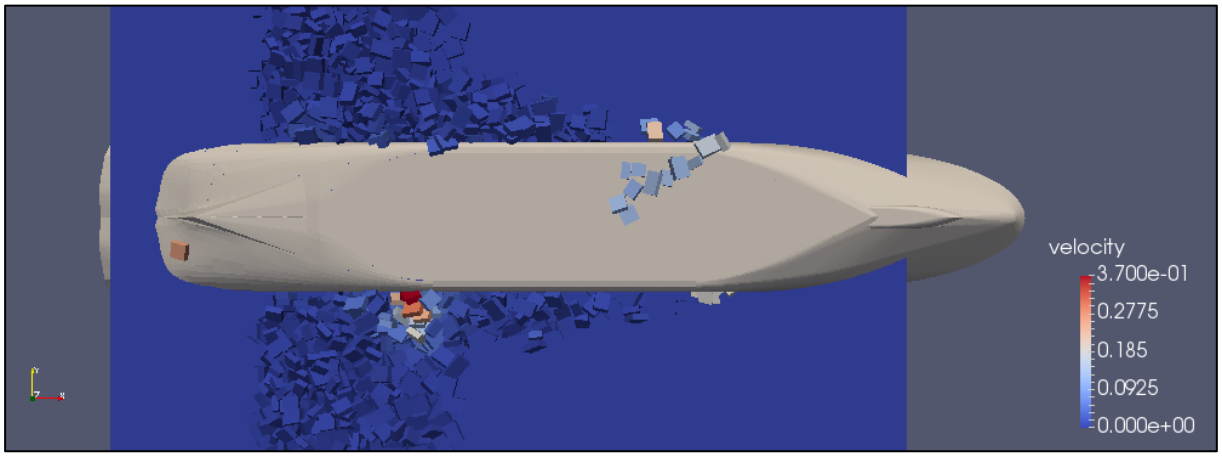

Figure 55. Bottom view, rectangular shaped rubble ice

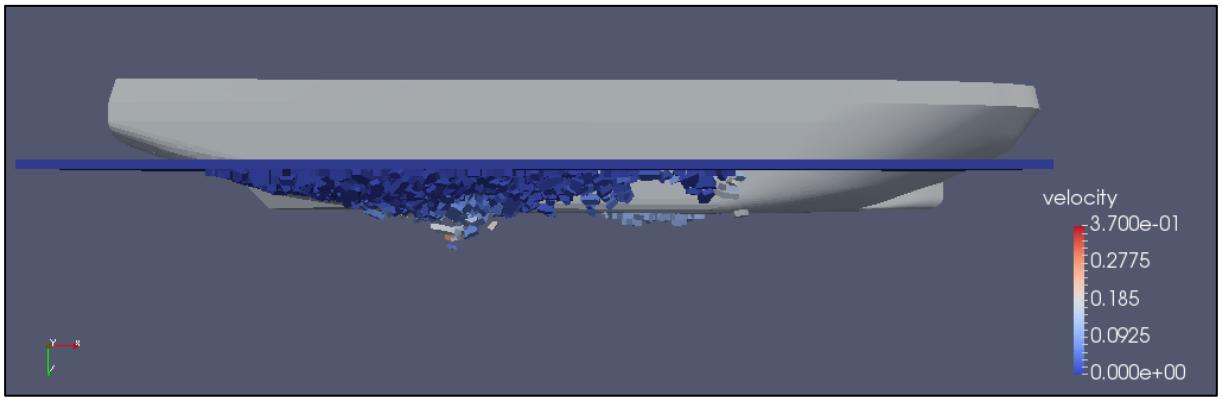

Figure 56. Side view, polygon shaped rubble ice

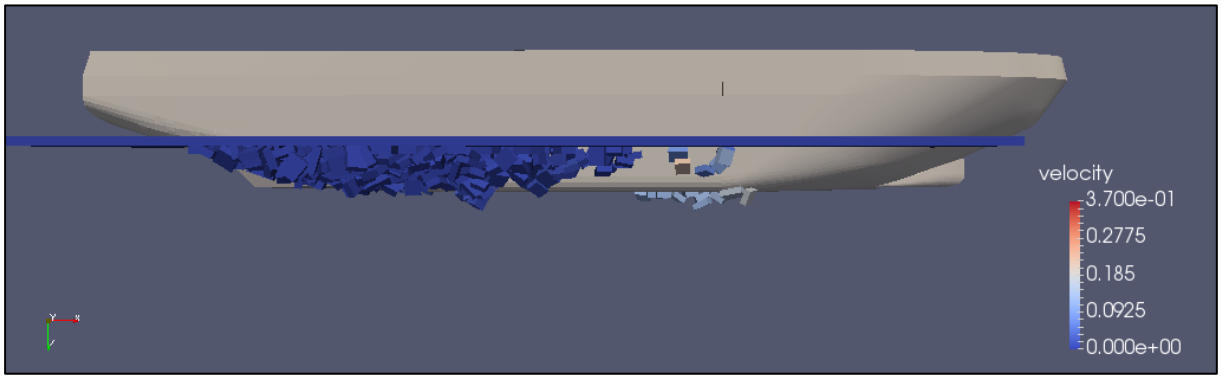

Figure 57. Side view, rectangular shaped rubble ice

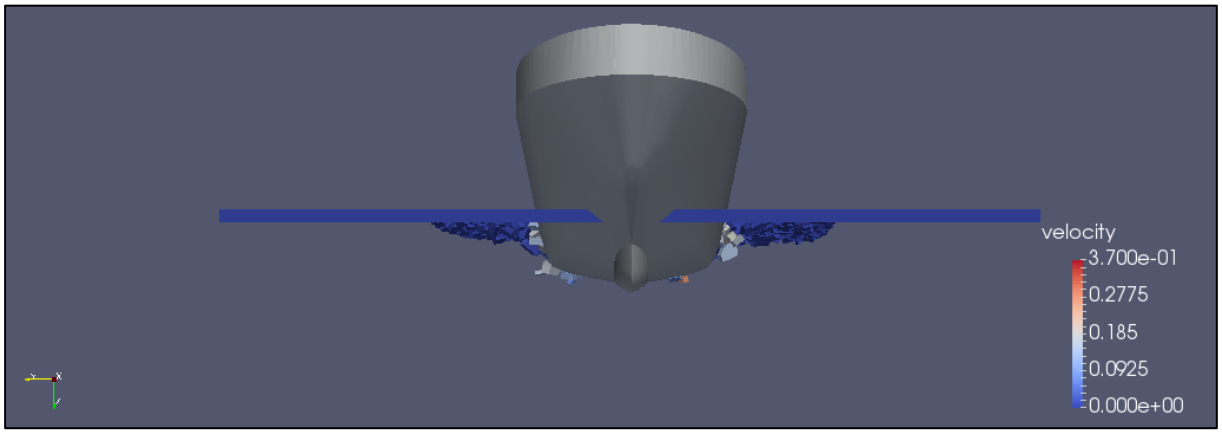

Figure 58. Front view, polygon shaped rubble ice

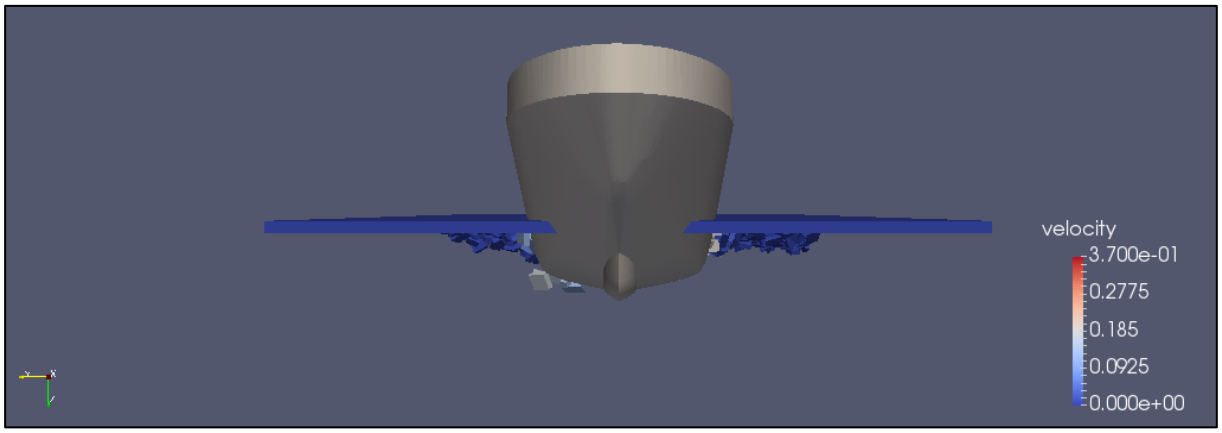

Figure 59. Front view, rectangular shaped rubble ice

5. CONCLUSIONS AND PROPOSALS

The numerical tool under development at HSVA is capable of simulating interaction between offshore structures and ships against ice ridge using the discrete element method (DEM). The purpose of the software is to perform simulations at the early design stage of the vessel.

This thesis introduces an implementation of polygon geometry to represent the rubble ice based on image processing of particles generated in ice ridges at HSVA ice basin. The data was processed using the open source software ImageJ. The main dimensions of the axis-aligned bounding box and the number of edges of the particles were analysed by beta probabilistic distribution functions and an algorithm to obtain random parameters based on the distribution was introduced in the code. The algorithm for the calculation of mass, volume, moment of inertia and centre of gravity of the polygons was implemented and validated.

Thereafter, simulations were performed using the code with rectangular and polygon shaped rubble ice for punch test and ship simulation. Both simulation's results were compared with experimental test under the same physical parameters to identify the performance of the codes.

The simulation of punch test showed that the implemented geometry modified the results significantly. The conclusion is that the punch simulation requires an improvement in the friction model. Currently, the Mohr-Coulomb friction model is used with a constant friction coefficient. Nonetheless, the freezing bonds of rubble ice are not modelled in the friction force calculation.

The ship simulation with the implemented feature showed positive results. The rubble ice shape does not influence the velocity profile of the ship, which was already satisfactory. The ice particles flow around the hull is significantly improved allowing the visualisation of particles towards the stern.

Overall, the computational time using polygon geometry of the rubble ice increased due to the mesh of the rubble ice elements. For future developments, the force calculation algorithm should be fully designed in parallel interface OpenMP to allow faster computation. In addition, the contact algorithm should be optimized for computational speed improvement. The last step to conclude the development of the software is the calibration of the force coefficients.

6. ACKNOWLEDGEMENTS

I would like to address my acknowledgements to Quentin Hisette for his close support throughout the development of the master thesis and internship at HSVA. Also, Peter Jochmann for advising the direction of this master thesis and supporting the development of this work.

To the colleagues at the Arctic department at HSVA a special thanks for sharing the expertise in ice technology.

I would like to thank the supervisor Prof. Bronsart, and the external reviewer Prof. Taczała. I extend the greetings to the EMSHIP professors from Université de Liège, École Centrale de Nantes and University of Rostock.

Finally, I would like to acknowledge the personal support from Ms. Yung Xin.

This thesis was developed in the frame of the European Master Course in “Integrated Advanced Ship Design” named “EMSHIP” for “European Education in Advanced Ship Design”, Ref.: 159652-1-2009-1-BE-ERA MUNDUS-EMMC.

7. REFERENCES

Alekseev, A., 2016. *Numerical Simulation of Ice Ridge Breaking*. Thesis (Master). University of Rostock.

Brown, B. and Lovato J., 2013. *General Random Number Generators* [online]. Florida State University. Available from: https://people.sc.fsu.edu/~jburkardt/f77_src/ranlib/ranlib.html [Accessed 17 November 2016].

Burkardt, J., 2005. *Geometric Calculations* [online]. Florida State University. Available from: https://people.sc.fsu.edu/~jburkardt/f_src/geometry/geometry.html [Accessed 17 November 2016].

Chen, J., 2012. *Discrete Element Method for 3D Simulations of Mechanical Systems of Non-Spherical Granular Materials*. Thesis (PhD). University of Electro-Communications.

Cheng, R.C.H., 1978. Generating Beta Variates with Nonintegral Shape Parameters. *Magazine Communications of the ACM*, 21(4), 317-322.

Ehle, D., 2011. *Analysis of Breaking through Sea Ice Ridges for Development of a Prediction Method*. Thesis (Bachelor), Universität Duisburg-Essesn.

Ehle, D., 2012. Ships Breaking through Sea Ice Ridges. *International Offshore and Polar Engineering Conference*, Rhodes, Greece, 1188-1193.

Ekeberg, O.C., Høyland, K. and Hansen, E., 2014. Ice ridge keel geometry and shape derived from one year of upward looking sonar data in the Fram Strait. *Cold Regions Science and Technology*, 109, 78-86.

ElSeify, M.O. and Brown, T.G., 2007. Keel-conical structures interaction: a parametric and sensitivity analysis of the shear-cap load model. *Recent Development of Offshore Engineering in Cold Regions*, Dalian, China.

Fenz, D.M., Foltz, R.R., Sarlis, A.A., Younan, A. H., Jochmann, P., Schroeder, C., Myland D., Ziemer, G. and Spencer, D., 2016. Case Studies in Application of Design of Experiments Methodology to Ice Basin Testing. *Offshore Technology Conference*. St. John, Canada.

Ferreira, T. and Rasband, W., 2012. *ImageJ User Guide* [online]. National Institute for Health. Available from: https://imagej.nih.gov/ij/docs/guide/ [Accessed 17 November 2016].

Hart, R., Cundall, P.A. and Lemos, J., 1988. Formulation of a Three-dimensional Distinct Element Model – Part II. Mechanical Calculations for Motion and Interaction of a System Composed of Many Polyhedral Block. *International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts*, 25(3), 117-125.

Hisette, Q., 2014. *Simulation of Ice Management Operations*. Thesis (Master). University of Rostock.

Høyland, K.V., Jensen, A., Liferov, P., Heinonen, J., Evers, K.U., Løset, S. and Määttänen M., 2001. Physical Modelling of First-Year Ice Ridges – Part I: Introduction, Consolidation and Physical Properties. *Proceedings of the 16th International Conference on Port and Ocean Engineering under Artic Conditions*. Ottawa, Canada.

HSVA, 2014. *Ice Ridge Preparation and Property Assessment*. Hamburg, Germany.

ISO/FDIS 19906, 2010. *Petroleum and natural gas industries – Arctic offshore structures, ISO TC 67/SC*. Final Draft International Standard. International Standardization Organization, Geneva, Switzerland.

Jensen, A., Løset, S., Høyland, K.V., Liferov, P., Heinonen, J., Evers, K.U. and Määttänen, M., 2001. Physical Modelling of First-Year Ice Ridges – Part II: Mechanical Properties. *Proceedings of the 16th International Conference on Port and Ocean Engineering under Artic Conditions*. Ottawa, Canada.

Lindqvist, G., 1989. A Straightforward Method for Calculation of Ice Resistance of Ships. *The 10th International Conference on Port and Ocean Engineering under Arctic Conditions.* Lulea, Sweden.

Martin, T., 2007. *Arctic Sea Ice Dynamics: Drift and Ridging in Numerical Models and Observations.* University of Bremen.

Matuttis, H.G. and Chen, J., 2014. *Understanding the Discrete Element Method.* Singapore: John Wiley & Sons.

Metrikin, I., 2014. A Software Framework for Simulating Stationkeeping of a Vessel in Discontinuous Ice. *Modeling, Identification and Control*, 35 (4), 211-248.

Polojärvi, A. and Tuhkuri, J., 2008. 3D discrete numerical modelling of ridge keel punch through test. *Cold Regions Science and Technology*, 59, 18-29.

Roerdink, J.B.T.M., Meijster, A., 2011. The Watershed Transformation: Definitions, Algorithms and Parallelization Strategies. *Fundamenta Informaticae*, 41 (2001), 187-228.

Seidel, J., 2016. *Numerical prediction of the keel resistance of first-year pressure ice ridges on offshore structures*. Thesis (Master). Technical University of Berlin.

Serré, N., 2011. Study of the rubble ice action in scale-model ice ridge impact on seabed structures. Thesis (PhD). Norwegian University of Science and Technology.

Tonon, F., 2004. Explicit Exact Formulas for the 3-D Tetrahedron Inertia Tensor in Terms of its Vertex Coordinates. *Journal of Mathematics and Statistics*, 1(1), 8-11.