Master's thesis

NTNU
Norwegian University of Science and Technology
Department of Civil and Environmental Engineering

Tuomas Avellan

Weight estimation of ice strengthened hull structures

Master's thesis in Cold Climate Engineering
Supervisor: Prof. Knut Høyland & Prof. Jukka Tuhkuri
August 2020

NTNU
Norwegian University of
Science and Technology

ELOMATIC
CONSULTING & ENGINEERING

Tuomas Avellan

Weight estimation of ice strengthened hull structures

Master’s thesis in Cold Climate Engineering
Supervisor: Prof. Knut Høyland & Prof. Jukka Tuhkuri
August 2020

Norwegian University of Science and Technology
Department of Civil and Environmental Engineering

NTNU
Norwegian University of
Science and Technology

Abstract

The prevailing trend in ship design has been shifting towards holistic design process. This means that the different design aspects have all been incorporated together in order to find the design that is successful in all set criteria. For this reason, the different design aspects have to be simplified and divided into multiple subsystems that are in relations through dependencies and parameters. One particular parameter which has a lot of dependencies, is ship weight. Weight estimation and particularly its minimization has always been one of the most important tasks in ship design process.

This thesis work studies the effects of structural design selections on steel weight of ship's ice strengthened hull structures. The research is conducted as a case study for a concept vessel designed to operate in first year ice conditions. Finnish-Swedish Ice Class Rules are chosen for this study.

A design and optimization tool was developed in order to create different structural designs for the pre-defined hull form and to calculate their minimum weight. The tool creates desired structure topologies based on user's commands and calculates the scantling requirements using the chosen ice class rules. Weight is calculated based on steel plate and profile selections. This selection is done using an optimization algorithm designed to solve constrained single-objective optimization problems with linear methods.

The tool calculated minimum weights for nine different structural designs with four different ice classes. Results indicated that the selected framing system has the biggest effect on weight. For the two greatest ice classes, frame spacing and ice stringer utilization also became moderately significant in terms of weight. The tool provides accurate and valuable weight data which can be used both in research and in the industry. This type of optimization and analysis could be used as a part of parametric ship design if it is developed further.

Preface

It has now been exactly six years since I started my university studies in Aalto. Time has flown by quickly but fortunately I'm left with unforgettable memories from all kinds of adventures. I have enjoyed this part of my life thoroughly but I'm also keen to move forward to face new challenges in lyfe.

I want to thank Elomatic and Antti Yrjänäinen for this great opportunity that I was given. I also want to thank my thesis supervisor Nikita Dementyev for his involvement and all the other colleagues who spared their precious time to guide me with my work. Special thanks are also in order for Professors Knut Høyland and Jukka Tuhkuri for all their help and guidance.

Finally, I would like to thank all my loved ones for supporting me, not only with my thesis work but also with all the challenges I've overcome. I cannot express my gratitude enough.

Otaniemi, Finland, 28.8.20208

Timo Tuomas Viktor Avellan

Contents

List of Figures

List of Tables

Abbreviations

CAD	Computer Aided Software
CFD	Computational Fluid Dynamics
FSICR	Finnish-Swedish Ice Class Rules
LIWL	Lower ice waterline
UIWL	Upper ice waterline
Traficom	Finnish Transport and Communications Agency
STA	Swedish Transport Agency
3D	Three dimensional

1 Introduction

1.1 Background for research

With the growing demand for better and more efficient ship design development, the naval architects face the increasing challenge of making as accurate estimates and optimal decisions as possible during the early ship design phases. Inaccuracies and deficiencies in concept and basic design phases will lead into more severe problems during later parts of the design process. As a consequence this can produce undesirable design compromises, costly redesigns, or significant performance cuts from the original shipbuilding contract. All of these scenarios have a negative impact for all the parties involved as the project expenses increase and potential performance of the vessel decreases.

To overcome these demands the prevailing ship design process has been slowly shifting towards a holistic design approach. This means that the design process is inherently coupled with design optimization, where the designer selects the best solution out of many feasible options on the basis of set criterion, or rather a set of criteria (Papanikolaou, 2010). Because ships are rather complex integrations of many different subsystems, the designers are facing a multi-objective optimization problem. This means that even the simplest components within a ship design have to be further simplified to be feasible for global optimization. Parametric design and optimization for the ship hull form was already introduced in 1998 by Harries and has been studied further ever since. Today the designers are able to generate multiple hull forms based on parametric optimization. All of these hull forms can be feasible for a set project, but they require further analysis so a proper selection can be conducted. This has created a demand for agile and flexible methods to design and evaluate multiple different aspects of the ship based on the hull form or other set criteria.

One of the most critical aspects for a good ship design is accurate weight calculation. It is vital for any ship design because almost every technical calculation is affected directly or indirectly by ships weight and its distribution. This is why uncertainties in weight calculation during early stages of design typically generate more design problems in comparison to errors in more advanced calculations (Rodríguez & Fernández, 2012). For example hydrostatics and hydrodynamics of a vessel are greatly affected by the underwater hull form which is determined by the draught of a ship. This draught is essentially controlled by the weight and the weight distribution of the ship. In addition, different loading conditions and thus structural requirements of a vessel are ultimately affected by the weight to some extent as well. To further emphasize the challenges in accurate and optimal decision making, it is important to understand that all of the selected designs have an effect on weight. This creates a spiraling optimization problem that is very sensitive to the continues design selections over the course of design process. In order to minimize this problem, two principal methods can be used:

- Collection of good weight data from references and its intelligent use during the design process
- Frequent iterations and updates of the weight calculations as more quality data becomes available

A more specific weight estimation and minimisation problem with case sensitivity can be found from the classification society rules and requirements for ships navigating in ice. Different classification societies have determined a variety of different ice classes for different ice conditions around the globe. These rules are made to ensure adequate strength and performance of hull, machinery, rudder, propeller and other steering arrangements in icy waters. This thesis work will focus more specifically on hull's structural requirements and their optimization by minimizing steel weight. In this thesis work steel weight is defined as weight of the steel hull, deckhousing, and superstructures, including plates, brackets, castings and welding material. A detailed definition for steel weight and the overall ship weight hierarchy is given in chapter 2.1.

A short study on the effects of Finnish-Swedish ice classes for hull's steel weight was published in 2007. The study compared the steel weight of a non ice class hull structure to the added weight from reinforcements required for different ice classes. The results indicated an increase in weight between 1 - 10 % showcased in table 1 (Alanko, 2007). Although Finnish-Swedish ice class classifications are mainly applicable for vessels operating in the Baltic, they are widely used as basis for many non-arctic ice class rules. It can be further argued that the steel weight increase for any arctic vessel designed to operate in ice would be even greater. This is because the highest Finnish-Swedish ice class (IA Super) is considered as an equivalent to the second lowest polar class (PC6) (Riska, 2019).

Ice Class	**Steel weight increase from reinforcements**
IA Super	8 - 10 %
IA	6 - 8 %
IB	3 - 4 %
IC	1 - 2 %

Table 1: Steel weight increases from structural reinforcements for Finnish-Swedish ice classes (Alanko, 2007).

Because steel weight is one of the largest ensembles considering the whole weight of the ship, steel weight minimization has always been an important goal for the designers. Ice induced forces are known to be one of the highest local forces acting on various locations around a ship hull. Thus these forces ultimately determine the strength requirements of local hull regions that can interact with ice. Reinforcements for ice induced forces are only one part of the complex structural design. Still they have a significant effect on steel weight as ice class rises higher. This raises the question of how much weight could be saved if some of the structural designs were optimized particularly for ice induced forces. This question becomes even more interesting when its coupled with the holistic design process. How much can parametric optimization be further improved if the designers are able to generate steel weight estimates accurately for all feasible hull forms, knowing that these estimates are based on weight optimised selections?

1.2 Research questions

The aim of this thesis work is to study the significance of different structural design options for steel weight of a ship. Particularly for ice strengthened hull structures. In this work, weights of ice strengthened hull structures from different structural designs are compared and analysed. A case study hull form with multiple ice classes is used to conduct this study. The structural designs differ from each other with different framing systems, varying frame spacing, ice stringer placing and with altering stiffener profiling. This work is conducted with an optimization program that is connected to a parametric 3D modeller software which contains the geometric hull model. Further discussion about the future development of this program coupled into the parametric design process will also be included. The research questions to be answered in this thesis are:

1. What are the weight differences of ice strengthened hull structures between different structural designs for the case study hull form?
2. How significant is this weight difference depending on the ice class?
3. Which design selections have the the greatest impact on weight?
4. How can this type of optimization and analysis be used in the future for parametric ship design?

1.3 Methods and restrictions

This study is conducted using a design and optimization tool developed with Python programming language and scripting environment in CAESES (3D geometric modelling software). The tool uses different functions in CAESES to dimension and analyse different structural elements for a case study hull form. As an input, the optimization tool requires a hull form and a set of ship particulars predetermined outside the scope of work. A library of available stiffener profiles and plate thicknesses are also defined for the tool.

After the setup process, the existing hull form is divided into different areas based on the chosen ice class rules. Multiple structural designs with different framing systems, ice stringer placements and varying frame spacing are fitted to this hull form within CAESES. Dimensions are calculated and transferred to the Python program, which then evaluates the requirements and selects appropriate plate and stiffening profiles using an optimizaion algorithm. This optimization aims to satisfy the requirements of selected ice class while trying to minimize steel weight. Output from the tool is the steel weight estimation for structural elements affected by the chosen ice class. These elements include:

- Shell plating within the ice belt area
- Ice stringers
- Framing system along the hull form

All of the designs and corresponding steel weights are saved so they can be further studied and compared against each other.

Other structural elements, such as: bulkheads, keels, girders, deck-, and superstructures are ruled out from the weight calculations as they are not directly affected by ice class requirements. For this reason, global load calculations and requirements on different ship sections are not included as they are heavily dependant on all of these structures. Weight distribution will also be left out of the scope of study. Effects on the performance levels nor production expenses won't be considered. The scope of study and optimization is strictly limited to weight minimization of ice strengthened hull structures.

The ice class rules chosen for this work are the Finnish-Swedish ice class rules. These rules are designed for vessels operating in the northern Baltic or areas with similar ice conditions. The reason for this selection is the fact that the case study vessel is a car ferry designed to operate in first year ice conditions. The profile of this vessel isn't suitable for arctic operations. Analysing the weights of arctic class reinforcements from different classification societies wouldn't be beneficial for this study.

1.4 Thesis Structure

After this introduction chapter, the theoretical background and detailed definitions required to understand the optimization process will be covered. More detailed descriptions will be given for ship weight hierarchy, hull's structural requirements, ice loading phenomena, ice class rules and optimization principles. The third chapter focuses on the optimization tool itself describing its functionality, methodology and listing all the different assumptions related to its use. After describing the optimization tool, the fourth chapter introduces the case study vessel, defines all the different structural designs used in the study and presents the calculated weight results. The fifth chapter will conclude the work, provide answers to the research questions and describe possible inaccuracies. Future development of this work will also be discussed in the fifth and final chapter.

2 Theory and definitions

In this chapter, the theoretical background is covered. The subjects included in this chapter are: ship weight, ship's structural requirements, ice loads, ice class rules, and optimization. The aim of this chapter is to introduce and further define different concepts which are required to better understand this thesis work.

2.1 Ship weight

Archimedes' principle defines the physical law of buoyancy as follows: A body immersed in fluid is subjected to an upward force equal to the weight of the fluid displaced. I.e. as ship floats, it displaces its own total weight of water. This total weight of a ship can be divided into two main hierarchies: Lightweight and deadweight of a ship. In this work, the definitions and subdivisions of these two hierarchies follow the established industry norm. Because the study subject outlined in chapter 1.3 is limited to specific hull structures, the focus is given for lightweight class subdivision. More specifically for steel weight. Similar definitions can be found in the works of Lewis (1988), Bertram (1998) as well as in the industry convention papers and encyclopedias, such as SOLAS (2020), Wärtsilä Encyclopedia of Marine Technology (2015).

Lightweight is defined as the mass of an empty vessel including the installed equipment essential for ships' normal operation. It is the displacement of a ship without cargo, crew, passengers, fuel, lubricating oil, ballast water, fresh water or any consumables. Because deadweight is the difference between displacement and lightweight, it includes all of these listed items. Deadweight is essentially a measure of ship's ability to carry various items on board. This basic weight division is presented in the following equation 1:

$$\Delta = TW = LW + DW, \tag{1}$$

where,

Δ	displacement
TW	total weight
LW	lightweight
DW	deadweight

Lightweight can be further divided into three parts in its simplest subdivision: steel weight of the hull, weight of the machinery, weight of the equipment and outfitting. This subdivision is presented in equation 2. As described briefly in chapter 1.1, steel weight is defined as the mass of a hull girder, deckhousing and superstructures. Steel weight consists from all the steel elements required to build these larger ensembles. Machinery weight consists from all the engine plant installations. Starting with the main engine machinery, batteries, shaft lines, propulsion units, and ending with all the auxiliary units within these systems. Equipment and outfitting weight is defined by all the remaining items and installations fitted to the vessel.

$$LW = W_s + W_m + W_{eo} \tag{2}$$

where,

W$_s$ steel weight
W$_m$ machinery weight
W$_{eo}$ equipment and outfitting weight

Weight calculations in this thesis are strictly related to steel weight. However, these calculations do not consider the entire steel weight but focus specifically on designated steel elements. Chapter 2.4.3 presents the structural elements affected by the ice class rules and within the scope of study. More specific list of restrictions and assumptions for this study are given in chapter 3.3.

2.2 Ship's structural requirements

The basic challenge faced by naval architects is to assess different loading conditions that act on a ship. There are various kinds of motions and interactions causing dynamic loading onto ship structures. Also static loading due to gravity and buoyancy forces have to be accounted for. These loads determine the structural requirements that have to be met in order for a ship to be classified as seaworthy. The structural design solution to fulfil these requirements is never unique. There are multiple different design approaches which can all be feasible and fulfil the set requirements. It is the job of a designer to evaluate these different designs in order to find the optimal one. All of the loading coming onto a ship structure on a seaway may be referred to as service loads. To assess appropriate responses to these service loads, four distinct strength criteria are used within the industry (Mandal, 2017):

- Longitudinal strength
- Transverse strength
- Torsional strength
- Local strength

As defined in chapter 1.3, the strength requirements considered in this thesis work are strictly limited to ice strengthening requirements. These requirements are primarily related to local strength and in some aspects to transverse strength due to the nature of ice loads. Longitudinal or torsional strengths are not typically on the limit from ice induced loading. However, longitudinal strength is an important factor for longer vessels and thus it greatly affects the selection of the framing system. For this reason longitudinal strength will also be defined as a concept in this work.

Local strength requirements assume significance in smaller areas where the hull structure can be subjected to very high loading. These requirements are typically higher in comparison to other strength requirements. High localized loads such as ice loads, cannot be overcome solely with global structural hierarchies which rely on good load transfer between different structural members. To mitigate deformation and avoid permanent damage, local strengthening is required. This can be accomplished with additional stiffening members like stringers in the side shell or increased number of frames.

Transverse strength is considered when the side of ship's hull is subjected to transverse loading. In the ice class rules this is mostly accounted with local strength requirements that define an ice belt structure around the hull form. It is still important to consider global ice loading when determining the strengths of larger structural elements like double side structures. Global loading from transverse ice compression can cause tremendous amount of damage simultaneously to the entire hull girder. Structural members like deck plates, side shell frames and web frames contribute towards transverse strength of a ship.

Longitudinal strength becomes an important factor as ship length increases. Due to the difference in weight and buoyancy distribution along the ship length, the hull girder experiences longitudinal bending moments. Just like in any slender beam structure simply supported from its edges, the maximum longitudinal bending moment is reached around the middle region. This maximum bending moment increases as the beam length increases. Ice loads themselves do not inflict high longitudinal bending moments for the hull girder unless the ship climbs partially on top of an ice ridge during ramming procedure. However, longitudinal strength requirements can become the most important requirements if the ship's length to breadth ratio is around 5 or higher. All longitudinal structures contribute towards longitudinal strength of a ship.

To satisfy either longitudinal- or transverse strength requirements, it is preferable to select the corresponding framing system for the vessel. However it important to understand that longitudinal framing does not protect the hull girder from local ice loads as well transverse framing does. This is because of the natur was explained in the introduction chapter 1.1. Also longitudinal framing does not contribute towards transverse strength.

2.3 Ice actions

In order to understand the principles of ice loads and their effects on ship structures, definitions pertaining to ice actions are introduced. Ship-ice interactions may consist of several different contact scenarios acting solely, or simultaneously around the ship hull. To design an appropriate vessel for ice covered waters, naval architects should consider all the possible interaction scenarios. Most common scenarios include (Kujala & Riska, 2010):

1. Ship's direct impact collision with thick ice
2. Ship's indirect impact collision with thick ice
3. Ship's advancement in level ice

4. Ship's advancement in ice field with ridges
5. Ship's advancement in brash ice
6. Ship's jamming in between two compressive ice fields

By analysing the individual loads and occurrence frequencies of all the possible contact scenarios, it is possible to define structural strength requirements with direct calculations. In order to model any of the ice loads and forces from these scenarios, the designer must further distinguish two important concepts and understand how they are used. First is the distinction between global and local forces acting on the ship hull. Second distinction is between average and maximum forces.

Local force refers to ice load that is either part of a single contact on a specified area or total load on any single structural element. For these reasons, local forces are most important when designing the strength requirements for local frames and plate panels. Global force refers to the total contact force throughout one single ice-ship interaction scenario, from the first impact to the last contact. Global forces can also refer to the sum of all the ice loads acting simultaneously on the ship hull. Global forces are important when determining the strengths of larger structural elements or evaluating the performance of a vessel in icy waters (Riska, 2019).

Maximum force determines the expected maximum contact force during one ship-ice interaction scenario. This maximum force is used for the strength analysis of various structural designs. The concept of average force refers to the time averaged force during the entire ship-ice interaction. This average force is most often used for calculating ship's resistance in ice to predict performance levels.

The focus of this thesis work is strictly given for strength analysis and therefore local maximum forces are the most important forces to consider. Although there are methods to asses all the different loads within each contact scenario, in reality ice reinforcement design process typically starts with predetermined loads which are specified in the chosen ice class rules. Ice class rules do not require specific load calculations for each contact scenario as the predetermined loads are based on the same principles. The chosen ice class rules are covered more thoroughly in chapter 2.4. In this chapter the fundamental principles affecting the ice loads are introduced. Chapter 2.3.1 defines the origin of ice loads, describes load limiting mechanisms, and explains how they are related to the most common failure modes of ice. More thorough definitions of local ice loads and forces are given in chapter 2.3.2. In-depth descriptions of all the various ship-ice interaction scenarios are not covered in this thesis.

2.3.1 Ice loads, load limiting mechanisms, and failure modes

Ice induced load onto a ship hull is a dynamic loading process that occurs whenever a ship comes into contact with an ice floe. This loading is the result of change in relative motion between a ship and an ice floe during a collision. The amount of energy within a dynamic ice load can thus be simplified to the following equation (Kujala & Riska 2010):

$$E_{collision} = E^0_{kin,ice} + E^0_{kin,ship} - E^1_{kin,ice} - E^1_{kin,ship} \tag{3}$$

where,

$E_{collision}$	Energy within a dynamic ice load
$E^0_{kin,ice}$	Kinetic energy of an ice floe before the collision
$E^1_{kin,ice}$	Kinetic energy of an ice floe after the collision
$E^0_{kin,ship}$	Kinetic energy of a ship before the collision
$E^1_{kin,ship}$	Kinetic energy of a ship after the collision

This equation ignores the fact that a ship can also gain potential energy by climbing on top of an ice floe. The thought behind this simplification is the fact that the pitching motion of a ship is most often only one small step in the entire collision process and thus the ship will return to its original stance at the end of it. In a ramming collision where the ship is stranded on top of an ice ridge for example, the potential energy has to be accounted for.

The dynamic ice load itself is limited by three possible mechanisms (Frederking, 1999):

1. Limit energy, or momentum
    - Ice feature hits a ship and the motions of both objects come to halt. The momentum (velocity times mass) of both colliding objects determine the load.
2. Limit force
    - Driving forces are not strong enough to fail the ice cover.
3. Limit stress
    - Driving forces are high enough that ice fails in contact with a ship. (This limiting mechanism yields the highest ice forces)

Because ice strengthened ships are designed to break ice features during normal operations and because this results in the highest ice forces, limit stress mechanism is always considered during the ship design process. In addition to the limit stress mechanism, it is important to understand that there are different ways ice can fail under stress. Ice pressures and forces applied onto the ship hull are varying constantly. The peaks of these stresses and forces occur just before the failure of ice or the ship hull itself. This is illustrated in figure 1 which presents a typical force-time graph measured from framing structures near the contact area between ice and ship hull. Because the peak forces occur just before the failure of ice, the failure mode of ice greatly affects the magnitude of ice forces applied onto the hull of a ship. Ice can fail in crushing, flaking, buckling (elastic-plastic compressive failure or creep failure), bending, and splitting via radial or circumferential cracking (flexural failure)

(Sanderson, 1988). The failure mode of ice depends on many different variables such as: Ice conditions, ice properties, loading speed, and shape of the contact surface. Most common failure modes of level ice during ship-ice interactions are breaking by bending and crushing.

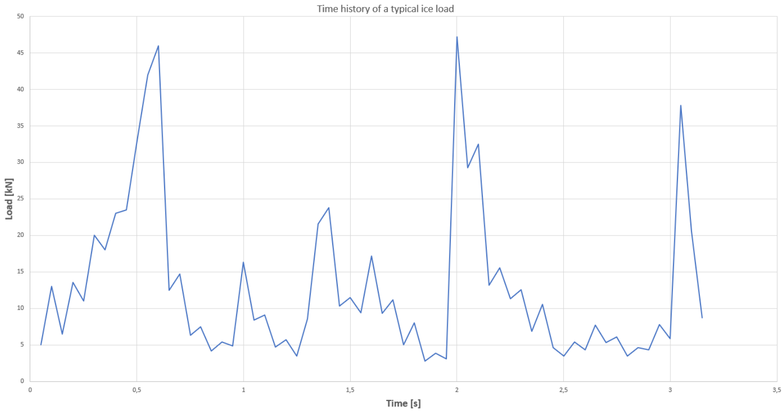


Figure 1: Example ice load measured from ship's framing structures during a ship ice interaction process. Measurements from JM Sisu are used as a reference (Kujala, 1994).

Most economical way for ships to break ice is to break it by bending the ice downwards (Riska, 2019). This is because the bending strength of sea ice is a lot lower compared to its compressive strength. Due to this high compressive strength, the highest forces present themselves at the local contact surfaces where crushing of ice occurs. Figure 2 gives a sideway view of a ship with a landing craft bow proceeding in level ice and breaking it by crushing and bending. The main forces are also shown in this figure. When the bow comes into contact with the ice edge, local crushing occurs on a small contact patch known as the nominal contact area. This nominal contact area represents the area where the hull and ice can be in direct contact with each other during the crushing process. The size of this area can be defined with geometries of the hull and the ice edge, and from the ship's penetration into the ice feature. As the ship continues to proceed forward the pressure and the force components increase. The vertical force component which is pushing the ice sheet downwards and the bow of a ship upwards, generates a vertical bending moment for the ice sheet. Eventually this bending moment exceeds the bending strength of ice and causes a bending crack to occur some distance away from the contact area. After the bending failure of ice, some of the built up pressure is released and the broken ice floe is submerged underneath the bow. Hydrodynamic and buoyancy forces pressurize the ice floe up against the hull as it is submerging along the hull form surface. This chain of events continues to repeat itself as the ship proceeds forward and starts crushing the new ice edge.

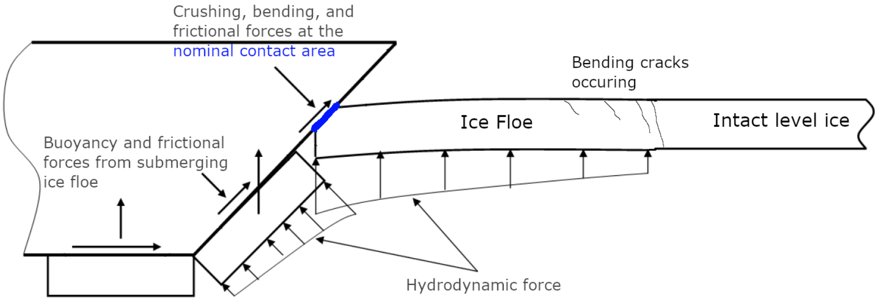


Figure 2: Forces in level ice which is broken by bending and crushing (Riska, 2019). Picture has been edited.

Because breaking by bending is the most economical way to break ice, it is also the most common way for ships to break ice. Therefore naval architects aim to design the over all hull form in a way that increases vertical bending forces and decreases horizontal crushing forces of ice. Crushing can never be avoided completely and thus it always has to be accounted for in the strength requirement calculations. Crushing of ice can happen wherever ice features come into contact with the hull form. This is taken into consideration within all the different ice class rules by defining different strength requirements for all the different hull regions that can come into contact with ice. This concept will be further explained in chapter 2.4. Knowing the most probable locations where crushing can occur is still an important aspect for the designer to grasp. This can help the designers to give special attention for local hot spots with severe crushing forces and to design a hull form which directs the broken ice pieces away from critical steering arrangements.

2.3.2 Local ice Loads

As defined earlier in chapter 2.3, local ice loads and more specifically local loads from crushing of ice produce the highest forces and pressures onto a ship hull. Therefore these high local pressures and forces from compressing ice determine the requirements for specific structural elements. During the late 1980's and 1990's, the concept of pressure-area effect was emerging within the international ice research community. The concept was made famous through a compilation from Sanderson (1988), which showed a pattern of decreasing ice pressure with increasing contact area. Two pressure-area relations related to this thesis work were termed by Frederking in 1998 and 1999 (Frederking, 1999): Process pressure-area relation, and spatial distribution pressure-area relation.

Process pressure-area relation describes the change in average pressures as a function of contact area during an impact collision. According to this theory, the average pressures during the entire collision process are higher for smaller areas. Measurements from Louis S. St. Laurent impact are presented in figure 3. In this figure the average pressures are plotted as functions of contact area sizes. The plotted points are connected with lines to indicate time sequence of data. A characteristic trend of decreasing pressure with increasing contact area can be seen.

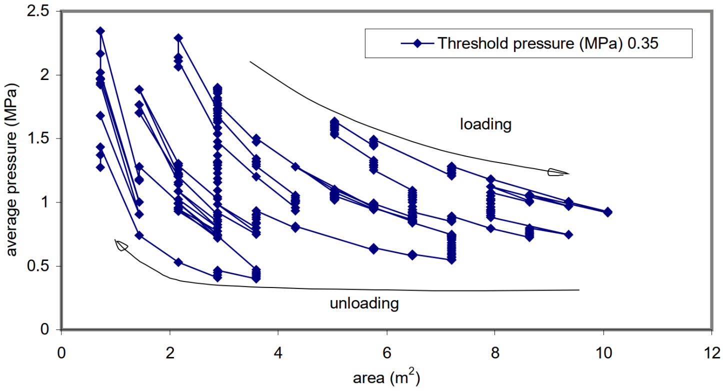


Figure 3: Average pressure as a function of contact area. Louis S. St. Laurent impact at 4,0 m/s against 1-2m thick ice (Frederking, 1999).

Spatial distribution pressure-area relation describes the average pressure on sub areas of various sizes within a larger area at one time instant. This theory also implicates that the average pressure is higher for smaller areas. Spatial distribution measurements from the same Louis S. St Laurent impact are presented in figure 4. Similar trend of decreasing average pressure with increasing area are can be observed. These measurements were also fitted with a best-fit pressure-area relationship graph which is often expressed with an equation 4:

$$p = C \times A^q \tag{4}$$

where, A is the studied contact area between a ship and ice. C and q are parameters describing ice properties, loading speed, aspect ratio, structure curvature, and local ice shape.

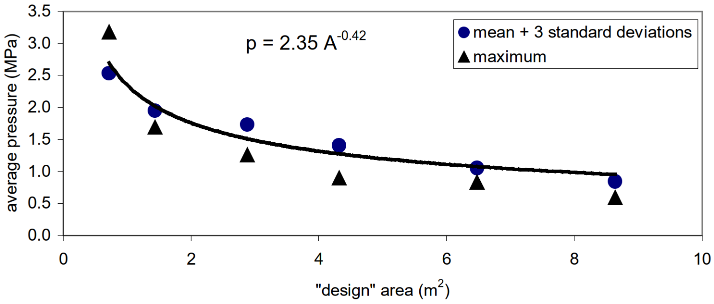
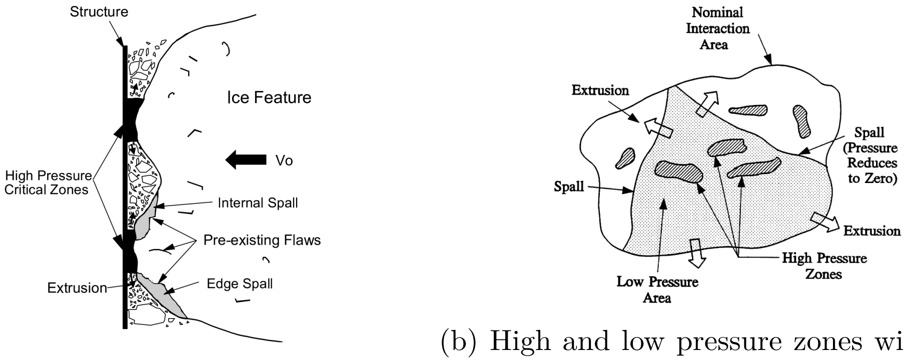


Figure 4: Average pressure as a function of contact area. Louis S. St. Laurent impact at 4,0 m/s against 1-2m thick ice (Frederking, 1999). Time instant is 2,07 seconds in figure 3.

Further studies on the ice crushing process from Ian J. Jordaan (2001), and from Daley, Riska and Tuhkuri (1998) have revealed that the actual crushing occurs on even smaller contact patches than the nominal contact area actually is. These so called high-pressure zones have been observed and measured to be more closer to point loads with diameters of only a few centimeters. Depending on the ice properties and geometries, these point loads tend to concentrate in specific ways forming very thin line loads along the contact surfaces. Detailed illustration of the crushing process onto a ship hull and the concentration of high-pressure zones are shown on figure 5. So called low-pressure zones can be observed to exist around the high pressure zones, where the crushed ice is extruded out of the nominal interaction area.



(a) Crushing of ice

(b) High and low pressure zones within the nominal contact area

Figure 5: Ice feature crushing against a ship hull from different views (Jordaan, 2001).

Today the typical determination process of local ice loads assumes that the nominal pressure from crushing is known for the nominal contact area. Smaller local design areas can then be defined freely somewhere within this area. Design pressures are then derived for these specific areas using the nominal pressure and other variables depending on the chosen ice class rules. For simplification purposes, these design pressures are typically set to be uniform across the corresponding design areas. The effects of high-pressure zones, low-pressure zones and other uncertainties related to the distribution of ice induced loads are considered within the derivation of design ice pressures.

2.4 Finnish-Swedish Ice Class Rules

Ice class rules are what define ship's requirements for operations in ice. Different classification societies have their own rules and guidelines to ensure sufficient safety and performance of ships operating in differing ice conditions. Depending on the chosen rules, the regulations can affect hull's structural design, engine and machinery systems, steering and propeller arrangements. Because the aim of this study is to evaluate weights of different ice strengthened hull structures, the rules affecting ship's hull are mainly concerned in this thesis.

The Finnish-Swedish Ice Class Rules also know as FSICR, are the chosen for this study. These rules have been developed by the Finnish Transport and Communications Agency (Traficom) and the Swedish Transport Agency (STA) in co-operation with classification societies. This development began in the early 1930s and the rules have been updated several times during the past years (Traficom, 2019). The rules are primarily intended for merchant ships which operate in the Northern Baltic during winter time. Ice conditions in the northern Baltic only include first-year ice. Due to the vast amount of field work with first-year ice and all the full-scale observations from ships navigating in the Northern Baltic, FSICR have become widely accepted by many different classification societies for vessels operating in any first-year ice conditions.

Design philosophy behind the Finnish-Swedish Ice Class Rules is partly built upon icebreaker assistance which is offered for vessels which meet the ice class requirements. The idea is to enforce efficient and safe operations for merchant ships in the Baltic sea. For economical reasons, excessive ice strengthening is desirably avoided and ships are required to have a minimum engine output in order to follow the assisting ice breakers or to maneuver in ice on their own. Different ice classes are categorised by the rules. Four of these ice classes are chosen for the study and their short descriptions by Traficom (2017) are listed below:

1. Ice class IA Super; ships with such a structure, engine output and other properties that they are normally capable of navigating in difficult ice conditions without the assistance of icebreakers;
2. Ice class IA; ships with such a structure, engine output and other properties that they are capable of navigating in difficult ice conditions, with the assistance of icebreakers when necessary
3. Ice class IB; ships with such a structure, engine output and other properties that they are capable of navigating in moderate ice conditions, with the assistance of icebreakers when necessary
4. Ice class IC; ships with such a structure, engine output and other properties that they are capable of navigating in light ice conditions, with the assistance of icebreakers when necessary

Information on the rule requirements and definitions which affect the ice strengthened hull structures are given in the following chapters. Further citations to the rules in these chapters are all taken from ice class regulation documents written by Traficom (2017).

2.4.1 Engine power

The regulations for minimum engine output are based on long term experience of Finnish and Swedish icebreaker assistance in the Baltic sea. As mentioned earlier, the underlying principle for winter navigation system is that all ships which meet the traffic regulations are given icebreaker assistance. An ice-classed ship is assisted by an icebreaker when the ship is stuck in ice or its speed has been substantially decreased by the ice conditions. The engine power requirements have been developed for navigation in brash ice channels in archipelago areas, at a minimum speed of 5 knots. The rules themselves do not guarantee that a ship is capable of navigating in thick level ice conditions or pushing through ice ridges without ice breaker assistance.

In the rules minimum engine output affects the ice pressure calculations and thus has an effect on ice strengthened hull structures. This is the reason for its inclusion into this study. The minimum engine output is defined as the total maximum output the propulsion machinery can continuously deliver to the propeller(s). If there are any machinery restrictions due to technical or regulatory reasons, the engine output shall be taken as the restricted output. If additional power sources are available for the propulsion, this power can be included into the total engine output. The required engine output is determined by a formula which accounts for the selected ice class, different particulars of the ship and its propulsion system. Detailed definitions and formulas used in the research work are presented in the appendix A.2.

2.4.2 Hull regions and vertical extensions of ice strengthening

Ice induced loads primarily occur at the waterline level where the ship initially comes into contact with floating ice, and below it where the broken ice pieces are submerged to. In addition, the ice loads on different locations on the hull also vary in magnitude because the contacts are very different depending on the location and shape of the hull. For these reasons the ice class rules divide the ship's hull into different regions and determine certain vertical extensions for the ice strengthened structures.

The Finnish-Swedish Ice Class Rules divide ship's hull into three different regions: Bow region, midbody region, and stern region. Upper and lower ice waterlines are also defined so different vertical extensions can be assigned correctly for each particular ship. The upper ice waterline (UIWL) is the envelope of the highest points of the waterlines which the ship is intended to operate in ice. The lower ice waterline (LIWL) is the envelope of the lowest points of the waterlines which the ship is intended to operate in. Figure 6 presents the different ice strengthened regions and both ice waterlines. Upper bow ice belt and forefoot regions are also displayed on this figure.

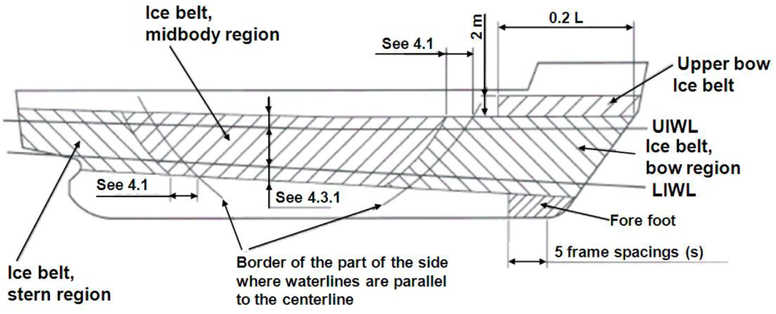


Figure 6: Ice strengthened regions of the hull according to FSICR (Traficom, 2017).

Based on these regions and ice waterline limits, the rules define vertical extensions for the ice strengthened shell plating (ice belt) and ice strengthened framing. These vertical extensions depend on the selected ice class. Furthermore all of the regions have different ice pressures which are used for scantling calculations of different structures within each region. Detailed definitions on region division and specific vertical extension limits are listed in the appendix A.4. They are used in the research work.

2.4.3 Structural requirements

The rules for structural requirements of a ship are related to local strength. Hull structures affected by these rules are: Shell plating, web frames, stringers, and frames. Their scantling requirements are determined with different design ice loads. These design ice loads are essentially empirical pressures, which are based on full-scale measurements taken from ships operating in the Baltic during winter time. The pressures do not reflect situations where a ship is stuck between compressing ice sheets and large ice forces are acting on the parallel midbody. It is assumed that icebreaker assistance is available if such an event occurs, leaving no time for serious damage to develop.

In the rules it is also assumed that ice pressures tend to act in a wave like manner, where pressure peaks occur at the framing structures. This phenomenon is illustrated in figure 7. This is because the contact between ice and the ship hull causes slight bending of plates which is greatest in between the supporting frames. The flexural stiffness of frames and shell plating are different. Due to this bending the contact pressure decreases in between frames.

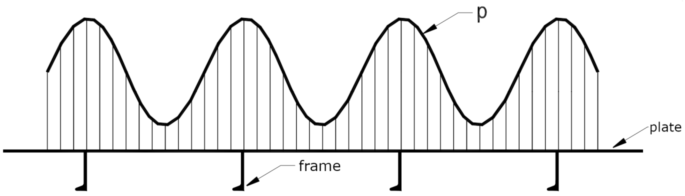


Figure 7: Pressure distribution on ship's side (Traficom, 2017). Picture has been edited.

Scantling requirements are determined for each structure type separately with different formulas. For shell plating the only scantling requirement is a thickness requirement that has to be fulfilled. Whereas all of the framing structures (web frames, stringers, and frames) have two scantling requirements: Shear area requirement and section modulus requirement. Shear area is a cross-sectional property of a structure member and is defined as the area of the section which is effective in resisting shear forces. Section modulus is a cross-sectional property as well and used to describe a structure member's ability to resist bending. Elastic section modulus is used within the rules because all the scantling requirements are based on elastic bending models.

Other notable assumptions and guidelines related to the scantling requirement calculations are listed below:

- Spacing distance and span length of a curved structure members are measured between two intersection points as a straight line. Figure 8 illustrates the span and spacing determinations for frames.

(a) Frame span

(b) Frame spacing

Figure 8: Definitions of frame span and frame spacing for curved members (Traficom, 2017).

- The effective breadth of the attached shell plate is to be used for calculating the combined section modulus for a framing structure. This means that when the section modulus requirement is determined for either a frame, a stringer, or a web frame, the plate has to be accounted for in the calculations.
- The calculated section modulus and shear area requirements have to be fulfilled in accordance with effective member cross-section. This means that if the supporting frame, stringer, or web frame is not normal to the plating, the cross-sectional properties have to be calculated using appropriate classification society rules.

All of the detailed definitions and formulas which are used to calculate ice pressures, required plate thicknesses, shear area- and elastic section modulus requirements, are presented in the appendix A. These formulas are used later in this research work.

2.5 Optimization

Optimization can be defined as a task of finding one or multiple solutions which correspond to minimizing (or maximizing) one or multiple objectives while satisfying all constraints (if any). A single-objective optimization problem involves a single objective function, whereas multi-objective optimization considers several conflicting objectives simultaneously. When all objective functions and constraints are linear functions, the problem can be called a linear problem. Optimization problems can however be nonlinear problems if at least some of the constraints or objectives are nonlinear functions. Solutions for these optimization problems can be either local or global. Local solutions are the best fit solutions from other nearby feasible options. Global solutions are the best fit solutions among all feasible options (Branke, et al. 2008).

Different optimization algorithms have been developed to help solving various problem types. In general, optimization algorithm can be defined as a set of instructions specifying how to start and conduct an optimization process until the final goal is achieved. Most of the time optimization algorithms have to impose the optimal solution before it is actually declared as the optimal solution. This is because the algorithm has to conclude that there is no better option. It is also important to understand that different algorithms most often fail to achieve the exact pre-specified goal, arriving instead to an approximation of the goal (Kishk & Mikki, 2008).

Before any optimization can be done, the problem must first be defined and modelled. This is as important or as critical as the optimization task itself. In this study, optimization is used for scantling selections. The optimization problem in question is a constrained single-objective problem. The objective is to minimize steel weight and the constraints are related to different scantling requirements. All of the constraints and the objective itself are linear functions, which makes the problem a linear one with only one solution. This optimization is performed with a simple optimization algorithm which is described in more detail in chapter 3.2.

3 Design and optimization tool

In order to conduct the case study for this thesis work, a design and optimization tool was developed to create and analyse different ice reinforced scantlings for any given hull form. Figure 9 presents an overview of the design and optimization tool and the information flow within it. The tool itself consist of two separate parts:

1. CAESES script (script coded for the 3D geometric modelling software CAESES)
2. Python program (executable program coded with Python 3.8.2 programming language)

The script within CAESES works as a structural designer for the chosen 3D hull form model. It generates line-, curve- and surface objects along the hull model based on the user's design input selection. These geometric objects are topological characterizations of different ice strengthened hull structures. All the relevant design information is then measured from each object and stored for further analysis. The script also measures additional ship hull related data which is required for the scantling calculations. All of this information can be transferred to external programs by saving it within a text file.

The Python program is used for calculating the scantling requirements and their minimum applicable weight for a given data set. It uses the previously mentioned text file as input and calculates the structural requirements of each separate object based on the Finnish-Swedish Ice Class Rules. The program then selects steel profiles and shell plates which fulfil the calculated requirements and have the lowest weight. This selection is done by utilizing a user defined library of different steel profiles and shell plates. As an output, the Python program exports an excel file containing all of the result data. This data includes all of the selected scantling profiles, their locations, spans or areas, and the calculated weights.

Both the CAESES script, and the Python program are covered more thoroughly in this chapter. Their methodology and specific functions as well as input and output formats are explained. Both of these tool parts can be used as standalone features.

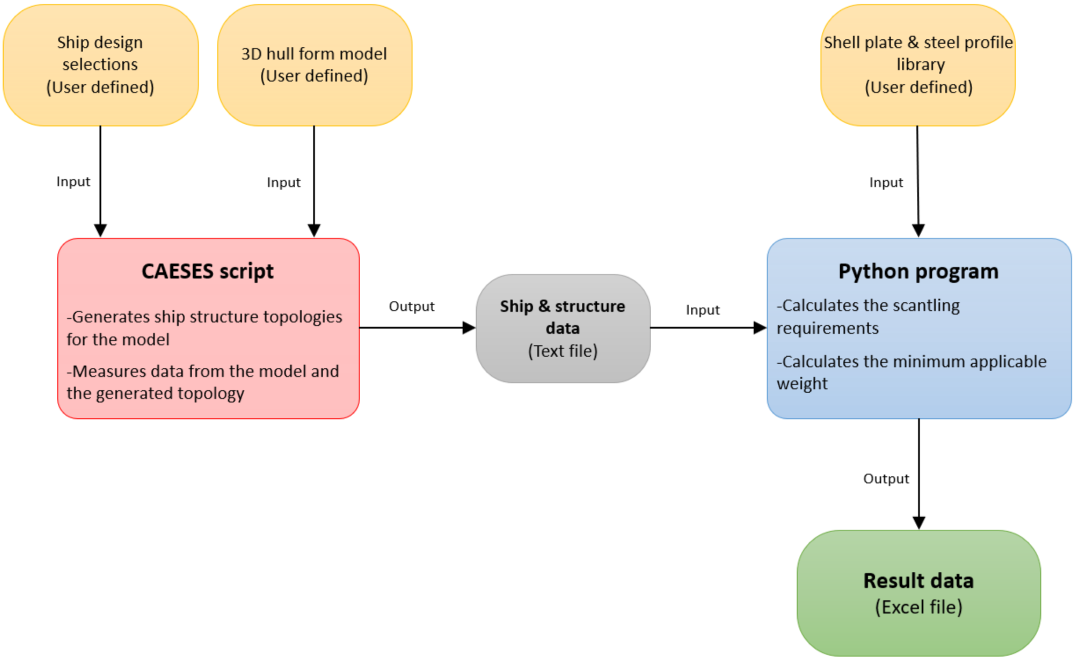


Figure 9: Design and optimization tool overview.

3.1 CAESES script

3.1.1 Overview and input

CAESES is a flexible CAD and CFD modeler for fast and robust designs. It is equipped with a variety of integrated optimization tools to analyze and explore different shape design options. Its foundation is a parametric 3D modeler. CAESES is designed as a command-driven platform which grants access to all of its functionalities through direct commands and scripts. CAESES and its scripting environment were used for this research work because they added a large degree of customizability and automation when performing tasks with a 3D hull form.

The CAESES script is designed to work as an interface for studying the case study hull form. Its primary objective is to obtain all the necessary information for ice reinforced scantling calculations. It has three primary tasks:

1. Calculating the minimum power requirement
2. Defining the hull region limitations for ice reinforcements
3. Creating hull structure objects and calculating required information for their scantling calculations

In order to perform these three tasks, the script requires a set of input variables which have to defined by the user. These input variables can be further divided into two categories: Control variables, and independent variables. Control variables are kept unchanged throughout the experiment whereas independent variables are controlled inputs varied during the experiment. All of the input variables are listed in table 2 together with brief definitions. A closer look on the specific values used within this case study experiment is given in chapter 4 where the case study ship is introduced.

Control variables	
Variable label	**Definition**
Hull model	3D ship hull model
LIWL level	Coordinate value to define lower ice water line level. Defined using Z-coordinate [m]
UIWL level	Coordinate value to define upper ice water line level. Defined using Z-coordinate [m]
Displacement at UIWL	Ship's displacement tonnage at UIWL [t]
Deck locations	List of coordinates to define deck locations. Defined using Z- & X-coordinates [m]
Bulkhead locations	List of coordinates to define bulkhead locations. Defined using Z- & X-coordinates [m]
Transverse plate floor spacing	Value to define the locations of transverse plate floors [m]
Web frame spacing	Value which is used to create web frames [m]
Controlled pitch propulsion	Definition whether the ship has controlled or fixed pitch propulsion [Yes / No]
N.o propellers	Value to define the number of propellers
Propeller diameter	Value to define the propeller diameter [m]
Independent variables	
Variable label	**Definition**
Ice class	Value to define the specific ice class [IC, IB, IA, IA Super]
Framing system	Definition of used framing system [Transverse / Longitudinal]
Frame spacing	Value which is used to create normal frames [m]
Ice frame spacing	Value which is used to create additional ice frames [m]
Stringer locations	List of coordinates to create ice stringers. Defined using Z- or Y-coordinates [m]

Table 2: List of different input variables that can be defined for the CAESES script

3.1.2 Script tasks and output

First task of the script is to calculate the minimum power requirement. As mentioned in chapter 2.4.1, the Finnish-Swedish Ice Class Rules have a minimum engine output requirement that has to be fulfilled by an ice class certified vessel. This power requirement affects the ice pressure calculations and is therefore needed for the scantling calculations. The CAESES script calculates this requirement using the information given in the inputs and by analysing hull line angles specified by the ice class rules. Formulas and definitions concerning the engine output calculations are presented in appendix A.2.

The script's second task is to define different hull region limitations for ice reinforcement calculations. As mentioned in chapter 2.4.2 and displayed on figure 6, the Finnish-Swedish Ice Class Rules first divide the hull into three parts: Bow, midbody, and stern regions. And further specify vertical extensions of ice strengthened shell plating (ice belt region), and ice strengthened framing for each region depending on the chosen ice class. Definitions on hull region division and the specified vertical extensions for ice strengthening are presented in appendix A.4. Figure 10 illustrates all the different hull region divisions and extension limitations applied onto a hull form using the script. The script does not define the upper bow ice belt or fore foot regions onto the hull form.

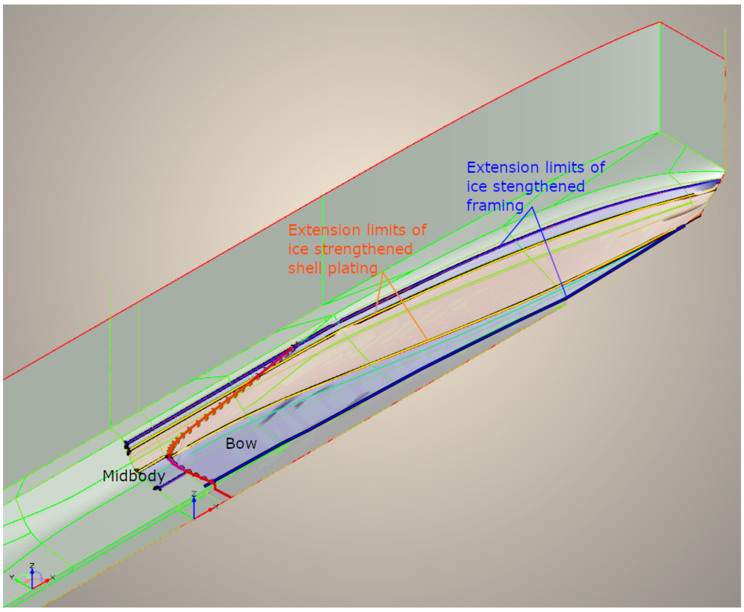


Figure 10: Snapshot from CAESES illustrating different hull region limitations.

The third and final task for the script is to create a topology of ice strengthened hull structures and measure all the required information for their scantling calculations. As defined in chapter 1.3 and 2.4.3, the structural elements included in this study are: Ice strengthened shell plates, -stringers, and -framing system. All of these structure types are automatically generated onto the existing hull form with dependencies to each other and to decks, bulkheads, and plate flooring structures defined outside the scope of study. These dependencies are related to the structural hierarchy of ship structures. This means that certain structure members cannot be created on top of each other and if two different structure types pass through each other, the structure with higher hierarchy status will cut the lower hierarchy structure into two parts. This results in a complex topology of different structures and enables the optimization of each individual structure object. Brief descriptions of creation methods and calculated information for each structure type are listed below:

- Shell plates are created as surface area objects on top of the hull form. Shell plating is generated using the previously defined hull regions, and vertical limitations as surface boundaries. This ice belt region is further divided into smaller subsurface areas by decks, and transverse bulkheads if they are located within the ice belt region. This division is done to create simplified strake lines, and seam connections for the ice strengthened shell plates. This enables the optimization of each individual shell plating area because the shell plate thickness doesn't have to be uniform along the entire hull form. An example surface patch is presented in figure 11 together with boundary illustrations. Information calculated from each surface patch is the area size.

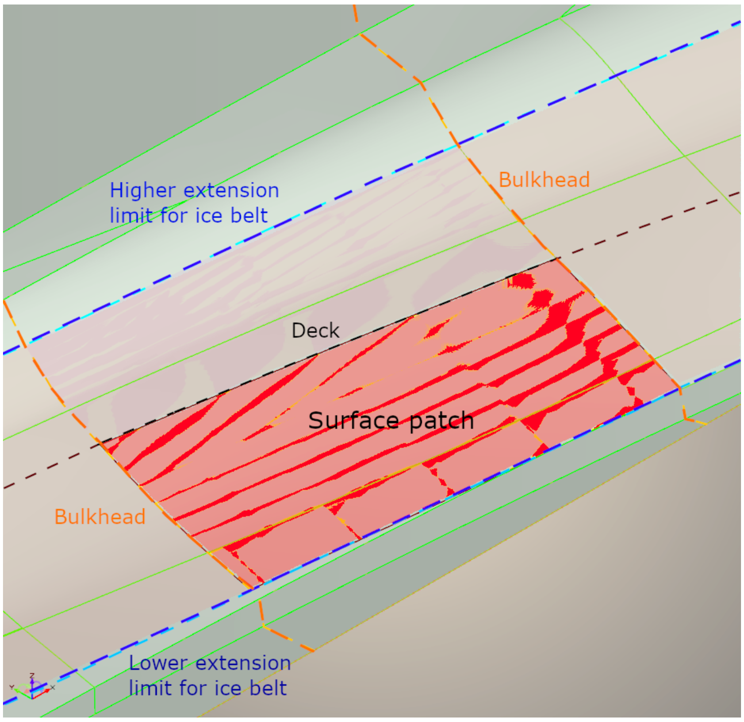


Figure 11: View from CAESES illustrating an example surface patch.

- Web frames are created as transverse surface curves following the hull form. They are generated along the ship length using the given web frame spacing. One web frame is essentially a cross-section of the hull surface created with a YZ-plane on the X-axis. Web frames extend from the bottom to the highest deck level defined. In case a transverse bulkhead, or a plate floor member already exists at the same location, web frame won't be created on top of them. Web frames can be split into multiple parts by longitudinal bulkheads or decks at cross points. Information measured from each web frame part is listed in table 3. An example web frame on a hull surface is presented in figure 12 together with illustrations on the calculated dimensions and boundary structures.

Information calculated from each web frame part		
Data label	**Values**	**Detailed definition**
Web frame's region	[Bow / Midbody / Stern]	-
Web frame within ice reinforced hull region	[Yes / No]	-
Index number of the attached shell plate area	[Index No.]	-
Web frame's true span	Span length [m]	Calculated along the hull surface. Used for weight calculation.
Web frame's span according to ice class rules	Span length [m]	Calculated as a straight line between start and end points. Used for scantling requirement calculations.
Web frame's spacing according to ice class rules	Spacing distance [m]	Calculated as a straight line to an adjacent structure (web frames or transverse bulkheads). Measured from mid span. From the two spacing measured (spacing on both sides of the web frame), the greater value is selected. Used for scantling requirement calculations.

Table 3: Six different values which are defined and calculated for each web frame part

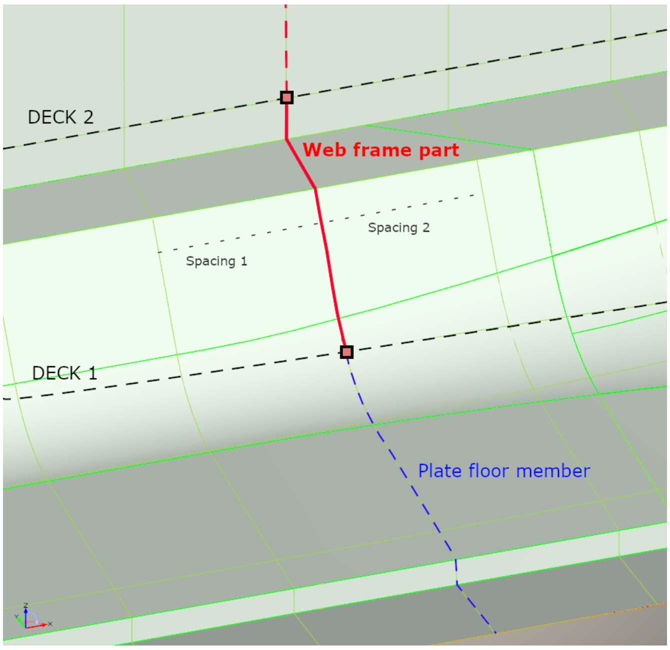


Figure 12: View from CAESES illustrating an example web frame part.

- Ice stringers are created as longitudinal surface curves following the hull form. They are generated on specific locations using the given coordinates. One ice stringer is essentially a cross-section of the hull surface created with a XZ-plane on the Y-axis or with a YX-plane on the Z-axis. Ice stringers start from a specified coordinate and extend towards the bow of the ship until coming into contact with a deck or the stem. Ice stringers can be split into multiple parts by bulkheads, and web frames at cross points. Information calculated from each ice stringer part is listed in table 4. An example ice stringer is presented in figure 13 with illustrations on the calculated dimensions.

Information calculated from each ice stringer part		
Data label	**Values**	**Detailed definition**
Ice stringer's region	[Bow / Midbody / Stern]	-
Index number of the attached shell plate area	[Index No.]	-
Ice stringer's true span	Span length [m]	Calculated along the hull surface. Used for weight calculation.
Ice stringer's span according to ice class rules	Span length [m]	Calculated as a straight line between start and end points. Used for scantling requirement calculations.
Ice stringer's spacing according to ice class rules	Spacing distance [m]	Calculated as a straight line to a nearby stringer or deck. Measured from mid span. Used for scantling requirement calculations.
Ice stringer's distance to the ice belt	Distance [m]	Calculated as a straight line to the nearest point within the ice belt area if the stringer is located outside the ice belt area. Measured from mid span. Used for scantling requirement calculations.

Table 4: Six different values which are defined and calculated for each ice stringer part

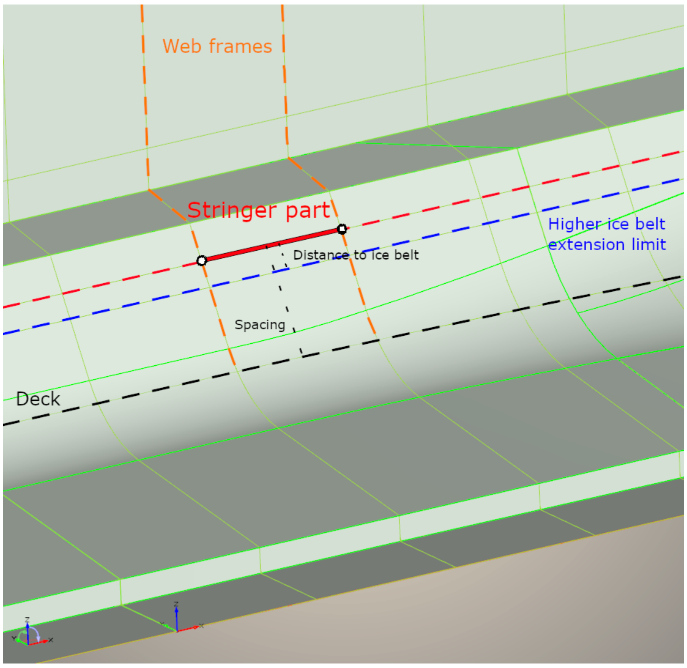


Figure 13: View from CAESES illustrating an example ice stringer part.

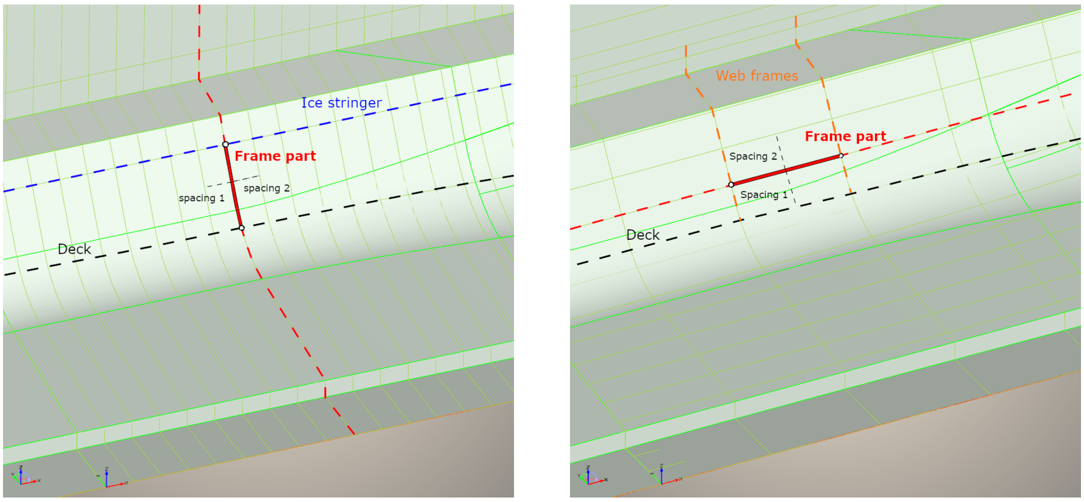
- Frames are created either as longitudinal or as transverse surface curves onto the hull form. They are generated along the ship length, height, or width depending on the chosen framing system and using the given frame spacing. One frame is essentially a cross-section of the hull surface created with a YZ-plane on the X-axis, a XZ-plane on the Y-axis, or a XY-plane on the Z-axis. Information calculated from each frame part is listed in table 5.

Information calculated from each frame part		
Data label	**Values**	**Detailed definition**
Frame's region	[Bow / Midbody / Stern]	-
Frame within ice reinforced hull region	[Yes / No]	-
Index number of the attached shell plate area	[Index No.]	-
Frame's true span	Span length [m]	Calculated along the hull surface. Used for weight calculation.
Frame's span according to ice class rules	Span length [m]	Calculated as a straight line between start and end points. Used for scantling requirement calculations.
Frame's spacing according to ice class rules	Spacing distance [m]	Calculated as a straight line to any adjacent structure (frames, web frames, stringers, bulkheads, decks, plating floors). Measured from mid span. From the two spacing measured (spacing on both sides of the frame), the greater value is selected. Used for scantling requirement calculations.

Table 5: Six different values which are defined and calculated for each frame part

  - Transverse frames are generated along the ship length using cross-sections on the X-axis. Transverse frames extend from the bottom to the highest deck level defined. In case a transverse bulkhead, a plate floor member, or a web frame already exists at the same location, frame won't be created on top of them. These frames can be split into multiple parts by longitudinal bulkheads, decks, or ice stringers at cross points. Additionally transverse ice frames can also be generated along the ship length using the given ice frame spacing. The only exception to their creation is that transverse ice frames extend from the highest deck level below ice strengthened shell plating to the lowest deck level above ice strengthened shell plating. An example transverse frame is presented in figure 14a with illustrations on the calculated dimensions.

  - Longitudinal frames located below the lowest deck are generated along the ship width using cross-sections on the Y-axis. Longitudinal frames located above the lowest deck are generated along the ship height using cross-sections on the Z-axis. Longitudinal frames extend towards the bow of ship until coming into contact with a deck or the stem. In case a deck, or a longitudinal bulkhead already exists at the same location, frame won't be created on top them. These frames can be split into multiple parts by bulkheads. An example longitudinal frame is presented in figure 14b with illustrations on the calculated dimensions.



(a) Transverse framing

(b) Longitudinal framing

Figure 14: Views from CAESES illustrating example frame parts

All the extracted information from each individual hull structure object is saved within a list format. Each structure type has its own list where the specific object lists are saved all together. As an output, the CAESES script prints out a text file which contains all the different lists and other information required for further analysis. The information printed to the text file includes:

1. Ice class
2. Framing system
3. Minimum engine power requirement
4. Ship displacement
5. List of shell plates
6. List of frames
7. List of ice stringers
8. List of web frames

3.2 Python program

Python is an open source programming language which can be used for a wide variety of applications. It is a popular general-purpose programming language due to its high flexibility and large library of third party modules. Python version 3.8.2 is used for this research work due to its flexibility and because the author has previous experience with the language.

The Python program is designed as a calculation software for ice strengthened ship structures. Its objective can be divided into three separate tasks:

1. Calculating the scantling requirements for given structure objects
2. Selecting a minimum weight scantling profile for each object using an optimization algorithm and a library of available steel profiles
3. Calculating the total weight of ice reinforced structures based on the selections

The scantling requirement calculations are performed using the FSICR introduced in chapter 2.4. The structure objects are given to the program as inputs. The scantling profile selection is based on steel weight and scantling requirements. As briefly explained in chapter 2.5, the optimization problem in question is a constrained single-objective problem with one optimal solution. The objective is to minimize steel weight while satisfying the calculated scantling requirements. This linear optimization is performed for each given structure object separately using an algorithm which selects the lightest scantling profile that fulfils the FSICR requirements. The profile selection is done from a pre-defined library of available steel profiles. After the selections, the program calculates the total weight of ice reinforced structures using the given dimensions and known weight properties of available steel profiles. As a result, the minimum total weight of ice reinforced structures is calculated for one structural design with a specified ice class. More detailed explanations of the programs functions and methodologies are given in the following chapters: 3.2.1, 3.2.2, and 3.2.3. The source code for the Python program can be found in the appendix E.

3.2.1 Input files

The program execution starts with parsing out information from two separate input files. First input is the text file introduced in chapter 3.1, which contains all the ship and structure related data from CAESES. This data is used for the ship scantling requirement calculations. Second input is an excel file containing information from all the available steel plates and -profiles. This data set is used for scantling selections and steel weight calculations.

Steel plates and -profiles within the excel database are divided onto three spreadsheets. Each spreadsheet contains a lists of different plate or profile elements with required properties for scantling and weight calculations. The three steel element categories are:

1. Shell plates
    - List of available shell plate thicknesses.
2. Bulb profiles
    - List of available bulb profiles and their properties. An example bulp profile and its properties are listed in table 6. Profile dimensions are illustrated on figure 15. Profile weight, neutral axis location (Cx), and second moment of inertia (Ix) are also given.

ID	**a**	**s**	**c**	**d**	**r**	**Area**	**Weight**	**Cx**	**Ix**
#	mm	mm	mm	mm	mm	cm^2	kg/m	mm	cm^4
HP100x5	100	5	15,5	14,8	4,5	6,75	5,3	61,3	67

Table 6: Bulp profile within the excel database

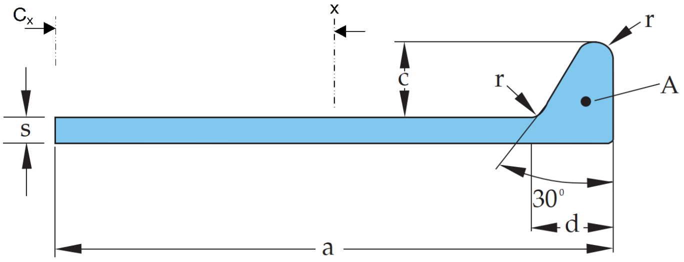


Figure 15: Bulp profile cross-section with dimensions

3. T-beam profiles

    - List of available T-beam profiles and their properties. An example T-beam profile and its properties is listed in table 7. Profile dimensions are illustrated on figure 16. Profile weight, neutral axis location (Cx), and second moment of inertia (Ix) are also given.

ID	**B**	**d**	**t**	**T**	**r**	**Area**	**Weight**	**Cx**	**Ix**
#	mm	mm	mm	mm	mm	cm^2	kg/m	mm	cm^4
200x140	142,20	201,50	6,80	11,20	10,20	29,30	23,00	151,30	1117,00

Table 7: T-beam profile within the excel database

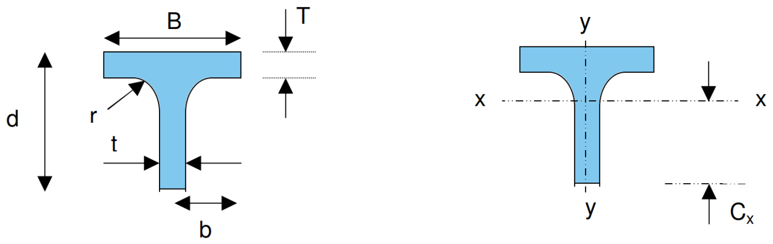


Figure 16: T-beam profile cross-sections with dimensions

All the parsed out information from both inputs is saved by the Python program. Complete plate and profile lists used in this study are attached to the appendix C. These profiles and their properties are taken from steel profile catalogues from Rautaruukki Oyj (2006) and Continental Steel Pte Ltd (2000).

3.2.2 Scantling calculations

Next step for the program is to perform scantling calculations for all structure objects saved from the input file. As explained in chapter 2.4, each structure type has its specific scantling requirement formulas which are presented in the appendix A.5. These formulas are coded into the program together with an optimization methodology to select the lightest appropriate scantling for each object. Brief descriptions of these methods are given below:

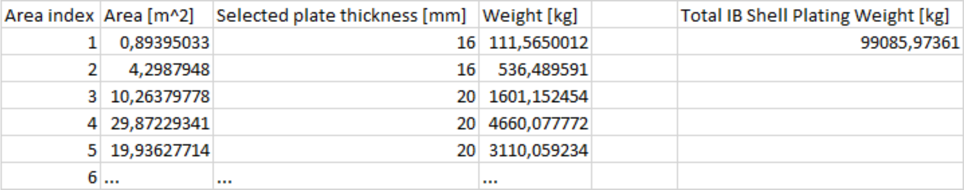
- Shell plating
    - Plate thickness is the only scantling requirement for ice strengthened shell plating. The required plate thickness is determined for each plate area separately by the program. In order to do this, the location and frame spacing information have to fetched for each plate area from the frame list. For each shell plate area, only the frame objects within the corresponding area are considered. The largest frame spacing within each area is used for the calculations. The thinnest plate thickness which fulfils the calculated requirement is then selected. Weight of each plate area is calculated using the known area size, chosen plate thickness, and steel density of 7800 kg/m^3.
- Framing structures (Frames, stringers, and web frames)
    - Shear area and section modulus are the two scantling requirements for ice strengthened framing structures. The requirements are determined separately for each structure part using the stored input data. When searching for steel profiles which fulfil the shear area requirement, only the cross-sectional areas of steel profiles are considered. Whereas effective breadth and thickness of the attached shell plate are used together with the steel profiles for combined section modulus calculations. Shell plate's effective breadth is determined to be the same as frame spacing or 650 mm at maximum. The thickness of the attached shell plate is known because plate thicknesses are already determined and because each framing structure object has an index number indicating which shell plate area it is attached onto. Equations used for the section modulus calculations for both bulb and T-beam profiles are presented in the appendix B. This appendix also includes figures highlighting the areas and dimensions which are used for these calculations.

      In case a structure part is not attached to any of the ice strengthened shell plating areas, the plate thickness used for the calculations is 16 mm by default. This thickness is taken from the case study ship's steel model introduced later in chapter 4.1. In case a structure part is not within the ice reinforced region limitations, the scantling calculations are performed using the lowest ice pressure in the rules. All of the structure parts are assumed to be perpendicular to the shell with fixed bracket supports on both ends. Bulb profiles are primarily considered for normal frames and intermediate ice frames. In case a suitable bulb profile cannot be found, larger T-beams are considered. For ice stringers and web frames, only T-beams are considered. Weight of each structure part is calculated using the known span length, and mass per metre property of the chosen steel profile.

3.2.3 Output file

After the program has completed all the scantling calculations, it creates an output Excel file for the results. These results are divided into five different spreadsheets containing the result data from specific structures.

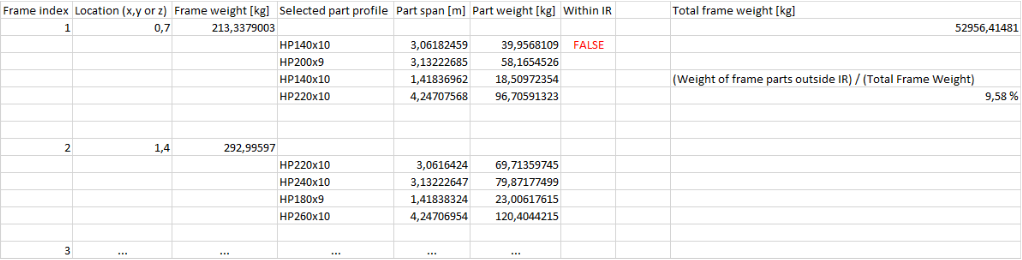
Shell plating results are printed on one spreadsheet. These results include area index numbers, area sizes, selected plate thicknesses, and calculated weights for each given plate patch area. The total weight of all shell plates is also calculated and included into the spreadsheet. Snapshot from an example result output is presented in figure 17.



Area index	Area [m^2]	Selected plate thickness [mm]	Weight [kg]		Total IB Shell Plating Weight [kg]
1	0,89395033	16	111,5650012		99085,97361
2	4,2987948	16	536,489591		
3	10,26379778	20	1601,152454		
4	29,87229341	20	4660,077772		
5	19,93627714	20	3110,059234		
6	...	...	...		

Figure 17: Picture from the output excel file displaying example results for shell plates

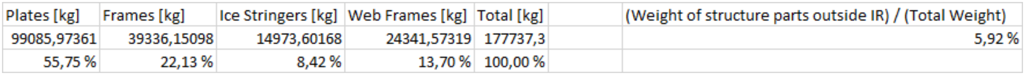
Framing structure results are printed on three separate spreadsheets. Spreadsheet division is between frames, stringers and web frames. The results include index numbers, location coordinates, and total structure weights. An additional list of selected profiles, profile spans, and calculated profile weights is also given for each structure. If a specific structure part is located outside of the ice reinforced region, it is marked with a "FALSE" status text. The total weight of all structures is also calculated and included into the spreadsheets. Snapshot from an example result output for frames is presented in figure 18. The result spreadsheets of stringers and web frames have the same formatting.



Frame index	Location (x,y or z)	Frame weight [kg]	Selected part profile	Part span [m]	Part weight [kg]	Within IR		Total frame weight [kg]
1	0,7	213,3379003						52956,41481
			HP140x10	3,06182459	39,9568109	FALSE		
			HP200x9	3,13222685	58,1654526			
			HP140x10	1,41836962	18,50972354			(Weight of frame parts outside IR) / (Total Frame Weight)
			HP220x10	4,24707568	96,70591323			9,58 %
2	1,4	292,99597						
			HP220x10	3,0616424	69,71359745			
			HP240x10	3,13222647	79,87177499			
			HP180x9	1,41838324	23,00617615			
			HP260x10	4,24706954	120,4044215			
3	...	...	...	...	...			

Figure 18: Picture from the output excel file displaying example results for frames

The fifth and final spreadsheet is an overview of the results. Total weights and their fractions from the over all total weight are presented for each structure type. The weight of framing structures outside the ice reinforced region divided with the over all total weight is also included. This is to indicate the weight proportion of structures which are calculated with a wrong set of rules. Snapshot from an example result overview is presented in figure 19.



Plates [kg]	Frames [kg]	Ice Stringers [kg]	Web Frames [kg]	Total [kg]		(Weight of structure parts outside IR) / (Total Weight)
99085,97361	39336,15098	14973,60168	24341,57319	177737,3		5,92 %
55,75 %	22,13 %	8,42 %	13,70 %	100,00 %		

Figure 19: Picture from the output excel file displaying example result overview

3.3 Assumptions and simplifications

Certain assumptions and simplifications are used within both parts of the design and optimization tool. These assumptions and simplifications should be acknowledged before analysing results from the case study.

- Upper bow ice belt or fore foot are not taken into account when determining the ice reinforced regions onto the hull form. Upper bow ice belt is only mandatory for IA and IA Super ice class vessels exceeding the speed of 18 knots. Ice strengthened fore foot is only required for ice class IA Super. The absence of these two additional regions make the weight calculations slightly non-conservative for IA and IA super ice classes.

- All of the framing structures generated onto the hull form terminate whenever crossing over a higher hierarchy structure. Each structure part is analysed independently. This allows detailed optimization but it might not reflect a realistic design if the profile of one singular framing structure is constantly changing.

- Ice strengthened shell plates, web frames, stringers, and frames are the only structures analysed by the tool. The scantling sizes of larger structures such as decks, bulkheads, plate flooring structures, and the keel are regulated by other classification society rules, thus assumed to be strong enough to withstand ice actions. Smaller elements such as brackets, collars or welds are not created for the model or accounted for in the weight calculations. The absence of smaller steel elements make the weight calculation slightly non-conservative.

- Weight calculations are based on calculated dimensions (areas and span lengths), and steel element properties taken from product catalogues within the steel industry.

- Finnish-Swedish Ice Class Rules are the only set of rules used in the analysis.
    - Shell plating outside the ice belt area is assumed to be same as in the reference steel model. This affects the scantling calculations of framing structures outside the ice belt area. This assumption can cause the weight calculation to be either conservative or non-conservative.

  - Scantling calculations for framing structures outside of the ice reinforced region are also done according to the ice class rules. The lowest ice pressure is used for these calculations. This simplification makes the weight calculations slightly conservative.

- All of the framing structure parts are assumed to have bracket supports on both ends. This means that fixed support on both ends can be used as the boundary condition for all section modulus requirement calculations.

- All the framing structures are assumed to be perpendicular to the shell plating. This makes the scantling calculations non-conservative because the possible angle of stiffening profiles is not accounted for in the shear area or section modulus calculations.

- The steel is assumed to be high-strength category. Yield stress σ_y used within the calculations is 315 N/mm^2.

- The effective breadth of shell plates is assumed to be the same as frame spacing or 650 mm at maximum.

4 Case study

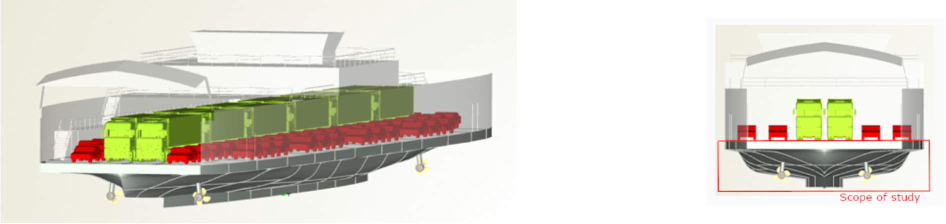
Case study for this thesis work is conducted with a concept design vessel designed by Elomatic Oy. Using the design and optimization tool introduced in chapter 3, nine different structural designs with four different ice classes are fitted onto the case study hull form and analyzed. This produces weight results for 36 different combinations. These results are further compared and evaluated to find answers for the research questions.

Overview on the case study vessel, and on the reference material is first given in this chapter. All of the structural designs chosen for this research are introduced next. At the end of this chapter, the weight result data is presented and analysed.

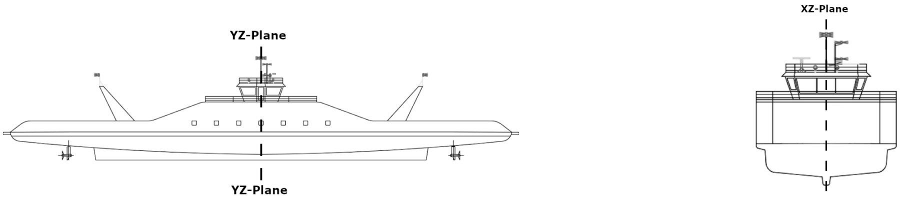
4.1 Holiship project

The case study vessel is a double ended car ferry designed to operate in first year ice conditions. The reference material used for this study includes a pre-defined hull form model, ship's current steel structure database with weight data, and ship's main particulars. Hull form area under the scope of study is limited between the bottom and the main deck (car deck). This is because ice reinforcement regulations are only affecting structures between these levels (on the Z-axis) due to the ship draft. Because the ship is double ended and the hull form has two planes of symmetry (XZ-plane, and YZ-plane), only one quarter of the hull form is required for the analysis. Pictures of the concept design and illustrations of the planes of symmetry are presented in figure 20.

A collection of ship's main particulars gathered from the reference material is presented in table 8. All of these main particulars are used as control variables for this study. As briefly explained in chapter 3.1, control variables are kept unchanged throughout the experiment. Main dimensions are bound to the parametric 3D modeler in CAESES and thus define the hull form model. Weight and propulsion variables are used for engine power requirement calculations. All of the structural arrangement variables are used to create boundary structures for the pre-defined hull form. They are also used to determine the different structural designs which are studied in this thesis work.



(a) Concept design (b) Scope of study



(c) YZ-plane of symmetry (d) XZ-plane of symmetry

Figure 20: Pictures of concept design vessel

Main particulars								
Main dimensions			**Weight**			**Structural arrangement**		
Length overall	*[m]*	122	Displacement at design waterline	*[t]*	1652,9	Deck 1 (center)	*Z-axis coord. [m]*	1,7
Length between perpendicualrs	*[m]*	84				Deck 1 (ends)	*Z-axis coord. [m]*	2,1
Breadth	*[m]*	19,2				Deck 2 (main car deck)	*Z-axis coord. [m]*	5
Design waterline	*[m]*	2,41				Web frame spacing	*[m]*	2,80
Depth	*[m]*	4,5						
			Propulsion			Longitudinal bulkheads	Transverse bulkheads	
			Num. Of propellers	*[-]*	4	*Y-axis coord. [+/- m]*	*X-axis coord. [+/- m]*	
			Propeller diameter	*[m]*	1,7	2,6	0	
			Controlled pitch propulsion	*[-]*	TRUE	5,7	5,6	
							14	
							22,4	
							33,6	
							44,8	
							50,4	

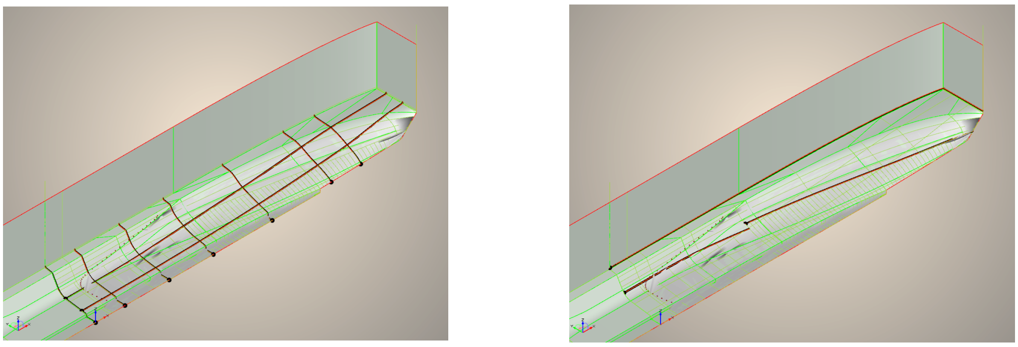
Table 8: Main particulars of the case study ship

4.2 Structural designs

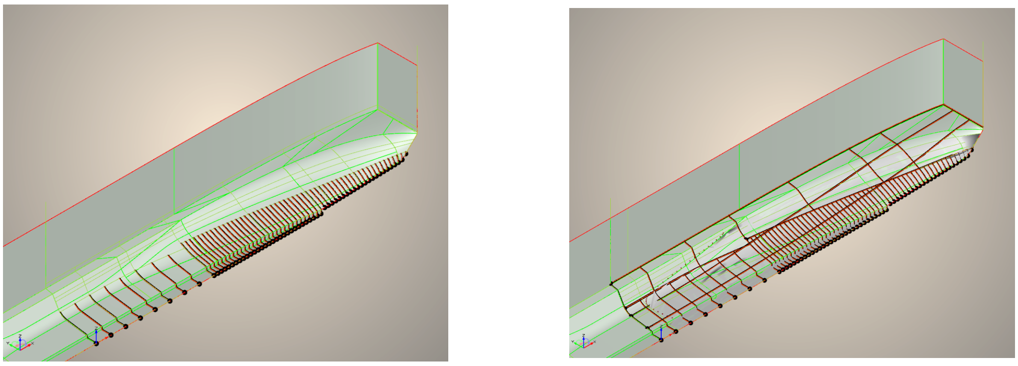
Each of the nine structural designs selected for this case study fit together with the pre-determined hull form and arrangement of primary structures. This is to maintain continuity of structures and to make sure each design is a realistic option to construct. The pre-determined primary structures include decks, bulkheads, and plate floor members. Their locations on the hull form are illustrated in figure 21.

Based on the locations of primary structures and hull region definitions, ice strengthened shell plating is divided into 12 separate shell plating areas. Ten of these areas are located in the bow and two in the midship. These areas are illustrated on figure 22. All of the structural designs have the same plate areas. The only difference comes with the varying ice class which determines the vertical extensions for the ice belt area.

The nine different case study designs can be divided into two categories: Transverse framing -, and longitudinal framing designs. Additional differences between the designs are the chosen frame spacing and ice stringer locations. Web frame spacing is the same for all designs. Web frames do not extend into the double bottom area because underneath deck one there is always a larger plate flooring member at the same location.



(a) Bulkheads (b) Decks



(c) Plate flooring members (d) Primary structures combined

Figure 21: Views from CAESES illustrating primary structures on the hull form model

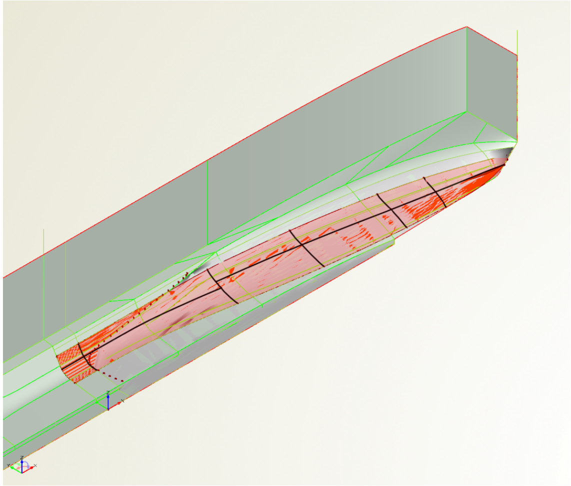
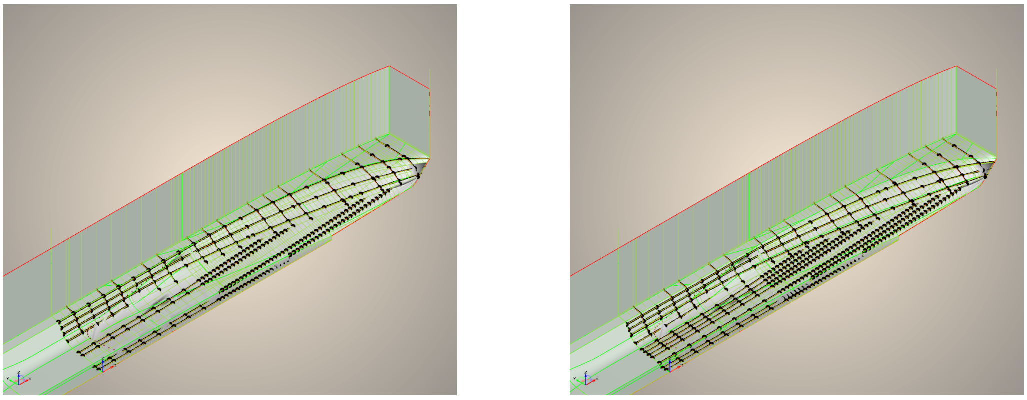


Figure 22: Shell plate area division for the case study

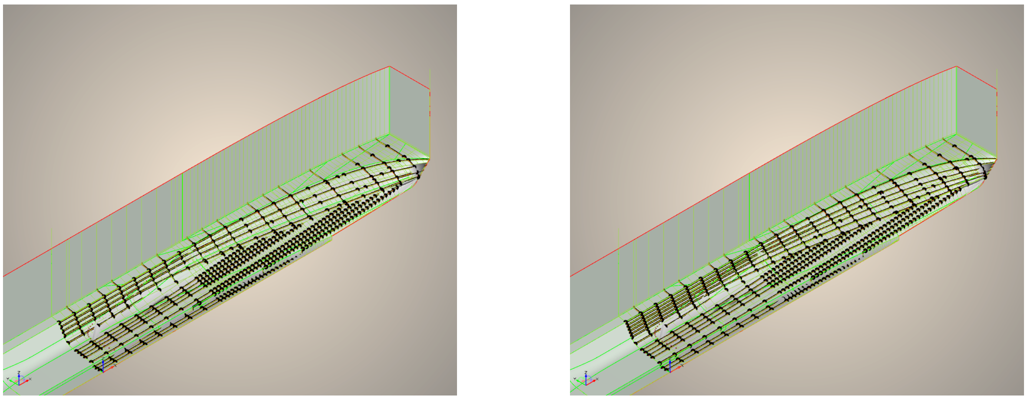
4.2.1 Longitudinal framing designs

Four different longitudinal framing system designs are described below. Pictures of all four designs are presented in figure 23.

1. Long 750
   - Longitudinal framing with 750 mm spacing.
2. Long 650
   - Longitudinal framing with 650 mm spacing.
3. Long 550
   - Longitudinal framing with 550 mm spacing.
4. Long 650 - 375
   - Longitudinal framing with varying frame spacing. Spacing of 650 mm is used underneath deck one. Spacing of 375 mm is used above deck one.



(a) Long 750 (b) Long 650



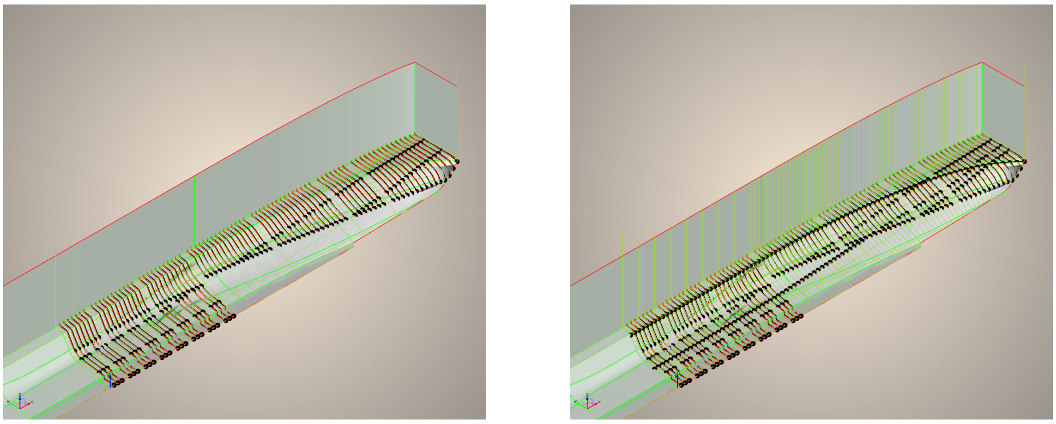
(c) Long 550 (d) Long 650 - 375

Figure 23: Longitudinal framing designs on the hull form model

4.2.2 Transverse framing designs

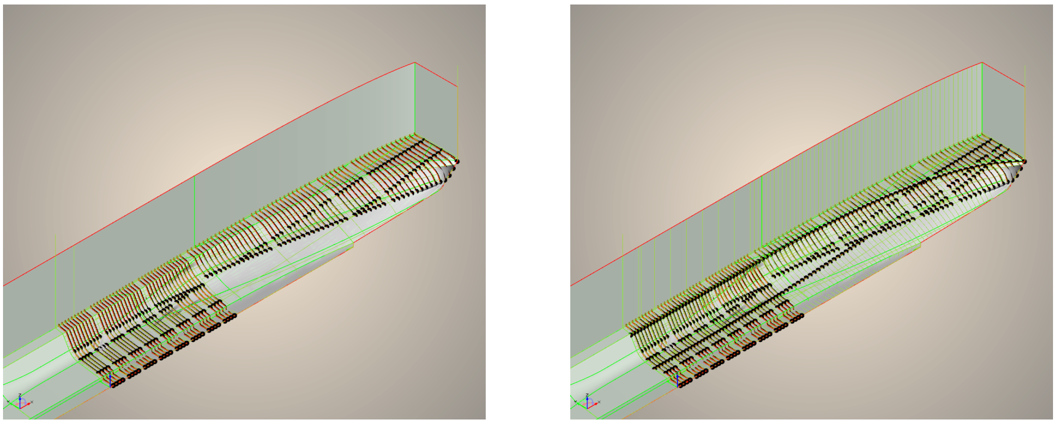
Five different transverse framing system designs are described below. Pictures of all five designs are presented in figure 24.

5. Trans 700
    - Transverse framing with 700 mm spacing.
6. Trans 700 & ice stringers
    - Transverse framing with 700 mm spacing and two ice stringers. First ice stringer is located underneath deck one in between the longitudinal bulkheads. It has a constant Y-coordinate and it extends until coming into contact with deck one. The first ice stringer is completely within ice reinforced region and partially within the ice belt region. Second ice stringer is located between decks one and two just above the ice reinforced region. It has a constant Z-coordinate and it extends to the stem.
7. Trans 560 - 700
    - Transverse framing with varying frame spacing. Spacing of 560 mm is used in between +/- 22.4 m on the X-axis. Spacing of 700 mm is used beyond these points.
8. Trans 560 - 700 & ice stringers
    - Transverse framing with varying frame spacing and two ice stringers. Spacing of 560 mm is used in between +/- 22.4 m on the X-axis. Spacing of 700 mm is used beyond these points. First ice stringer is located underneath deck one in between the longitudinal bulkheads. It has a constant Y-coordinate and it extends until coming into contact with deck one. The first ice stringer is completely within ice reinforced region and partially within the ice belt region. Second ice stringer is located between decks one and two just above the ice reinforced region. It has a constant Z-coordinate and it extends to the stem.
9. Trans 350mm & ice stringers
    - Transverse framing with 350 mm and two ice stringers. Normal frames have a spacing of 700 mm and intermediate ice frames have a spacing of 350mm. First ice stringer is located just underneath the ice belt region and second is located just above the ice reinforced region. Both ice stringers have a constant Z-coordinate and they extend to them stem.



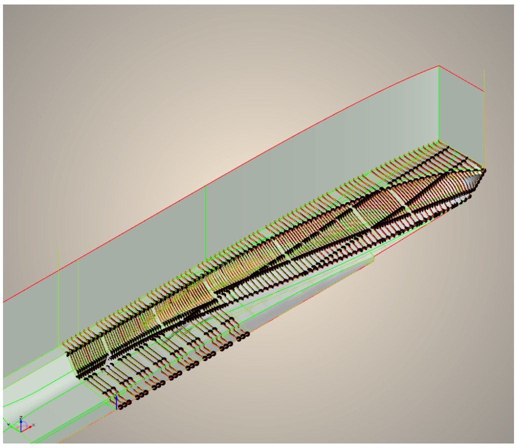
(a) Trans 700

(b) Trans 700 & ice stringers



(c) Trans 560 - 700

(d) Trans 560 - 700 & ice stringers



(e) Trans 350 & ice stringers

Figure 24: Transverse framing designs on the hull form model

4.3 Results

Analyzed results for each of the four ice classes are presented separately in the following chapters. All weight results from the design and optimization tool are multiplied by four because they were measured from one quarter of the hull form. Weight results for individual ice class and design combinations are listed in the appendix D.

Weight data from the initial steel model is also used in the analysis to get rough steel weight estimations for the entire ship. Corresponding plate areas and framing structures that are generated onto the hull form by the tool are stripped away from the steel model. This way a steel weight of structures outside the scope of study can be calculated for each ice class. By adding the weight of optimized ice reinforced structures, an estimate of the total steel weight can also be calculated. The steel model has a longitudinal framing system with 650 mm frame spacing. For this reason the combined estimates for ship's total steel weight are not entirely accurate, especially for transverse framing designs.

4.3.1 IC

Weight results for ice class IC are presented in figure 25. The lightest structural design is transverse framing with 700 mm frame spacing. Weight differences to other transversely framed designs are between 1,8 % and 4,4 % which is not significant. All of the longitudinal designs are heavier in comparison to transversely framed options. Heaviest of the designs is longitudinal framing with 650 - 375 mm spacing which has a 12,3 % weight difference to the lightest design.

The biggest difference between transversely and longitudinally framed designs is the weight from ice strengthened shell plating. Shell plating is the largest weight contributor with 44,0 - 68,5 % proportion from the total weight of ice reinforced structures. Weights from plates and web frames are quite similar for designs using the same framing system. The differences between designs using the same framing system are more related to normal frame weight.

For Transverse framing designs without ice stringers, frames are 29 - 32 % of the total ice reinforced structure weight. If ice stringers are used, weights of the stringers are between 8,5 % and 10,1 % of the total weight. Weight difference between designs using the same frame spacing but including or not including ice stringers is non existent. The design with intermediate ice frames differs from the other transverse designs with significantly lower weight from plates. However the weight from added frames makes it the heaviest transverse framing design.

For longitudinal framing designs the weight increases as the frame spacing decreases. The frame weight varies between 18,9 % and 28,8 % from the total ice reinforced structure weight. There are small differences between the selected plate thicknesses but it is not enough to counter the added weight from increased number of frames.

The steel weight of structures outside the scope of study is calculated to be 1245 tonne for ice class IC. By rough estimates for the entire ship, this means that the steel weight increase from the lightest structural design would be 0,22 - 1,48 % between all the other design options.

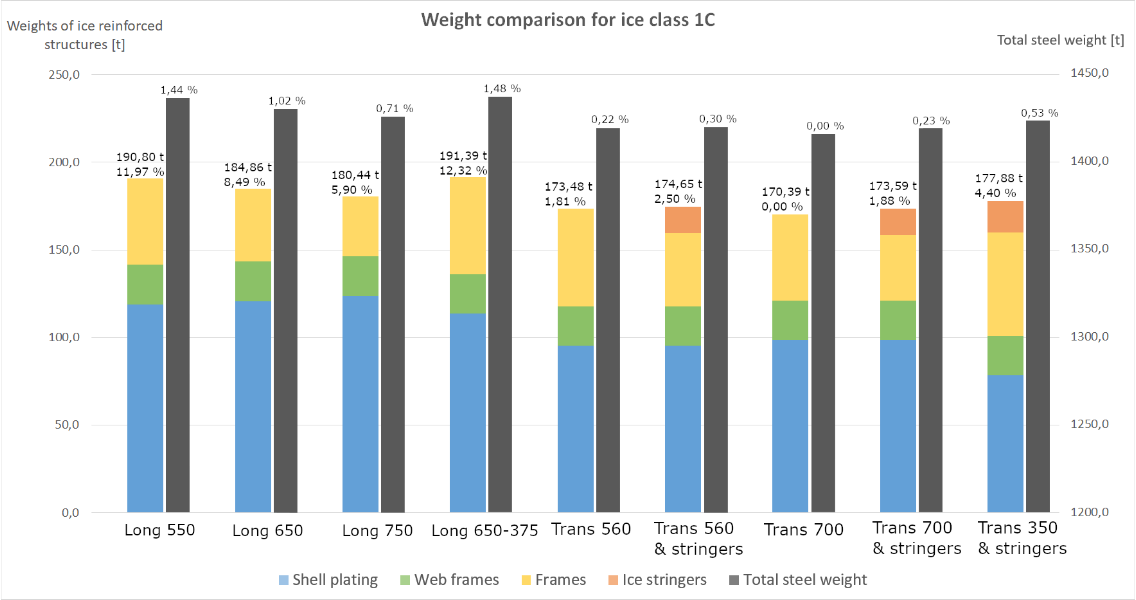


Figure 25: Weight results for ice class IC. Weight increase percentiles from the lightweight option are written above weights of ice reinforced structures and the total steel weight correspondingly.

4.3.2 IB

Weight results for ice class IB are presented in figure 26. Overall the weight results are very similar in comparison to the results from IC. Ice reinforced structure weights increase by 5 - 12 tonne from the corresponding weights of IC class.

The lightest structural design is transverse framing with 700 mm frame spacing. Weight differences to other transversely framed designs are between 0,8 % and 3,5 % which is not significant. All of the longitudinal designs are heavier in comparison to transversely framed options. Heaviest of the designs is longitudinal framing with 550 mm spacing which has a 14,6 % weight difference to the lightest design.

Biggest difference between transversely and longitudinally framed designs is the weight from ice strengthened shell plating. Shell plating is the largest weight contributor with 43,1 - 67,6 % proportion from the total weight of ice reinforced structures. Weights from plates and web frames are quite similar for designs using the same framing system. The differences between designs using the same framing system are more related to normal frame weight.

For Transverse framing designs without ice stringers, frames are 30 - 33 % of the total ice reinforced structure weight. If ice stringers are used, weights of the stringers are between 13,3 % and 13,7 % of the total weight. Weight difference between designs using the same frame spacing but including or not including ice stringers is non existent. The design with intermediate ice frames differs from the other transverse designs with significantly lower weight from plates. However the weight from added frames makes it the heaviest transverse framing design by a small margin.

For longitudinal framing designs the weight increases as the frame spacing decreases. The frame weight varies between 19,2 % and 29,2 % from the total ice reinforced structure weight. There are small differences between the selected plate thicknesses but it is not enough to counter the added weight from increased number of frames.

The steel weight of structures outside the scope of study is calculated to be 1245 tonne for ice class IB as well. By rough estimates for the entire ship, this means that the steel weight increase from the lightest structural design would be 0,10 - 1,81 % between all the other design options.

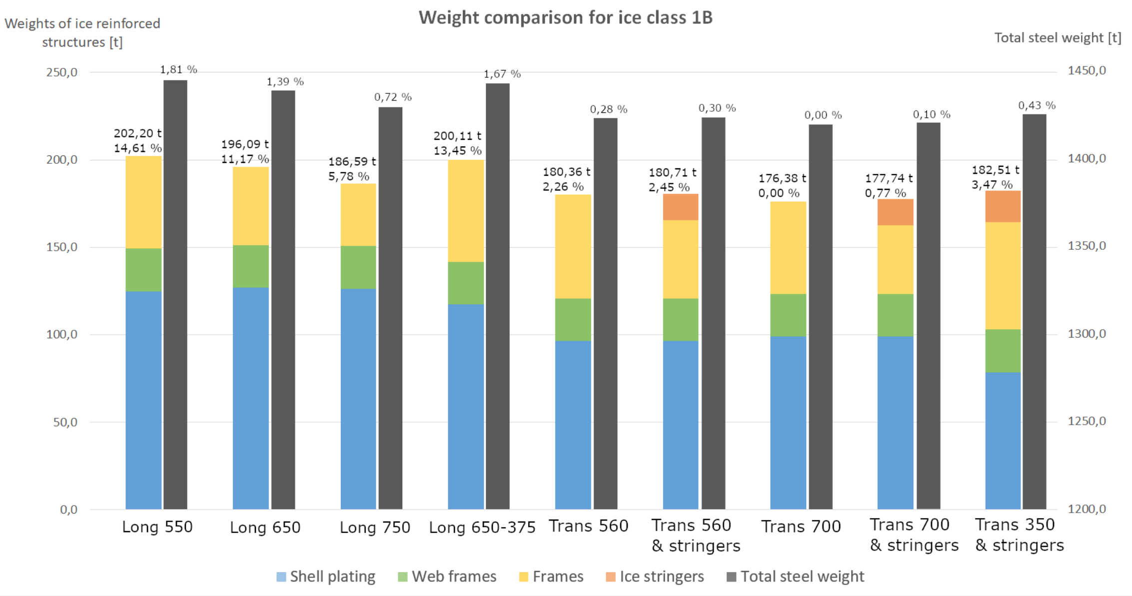


Figure 26: Weight results for ice class IB. Weight increase percentiles from the lightweight option are written above weights of ice reinforced structures and the total steel weight correspondingly.

4.3.3 IA

Weight results for ice class IA are presented in figure 27. Some variance from the results of IC and IB classes may be noted. Weights of transverse framing designs increase by 27 - 45 tonne from the corresponding weights of IB class. Weights of longitudinal designs increase by 50 - 60 tonne with a similar comparison. Overall more drastic weight increase can be seen from ice class IB to IA than there was from IC to IB.

The lightest structural design is transverse framing with two ice stringers and 350 mm frame spacing. Weight differences to other transversely framed designs are between 5,0 % and 7,4 %. All of the longitudinal designs are heavier in comparison to transversely framed options. Heaviest of the designs is longitudinal framing with 550 mm spacing which has a 25,1 % weight difference to the lightest design.

Weight of ice strengthened shell plating is still the biggest difference between transversely and longitudinally framed designs. The shell plating accounts for 46,2 - 69,7 % of the total ice strengthened structure weight. The significance of it is even greater for IA class than it is for IB and IC classes. Web frame profiles and shell plating thicknesses are still relatively similar for designs using the same framing system. The differences between designs using the same framing system are still related to normal frame weight.

Transverse framing designs with ice stringers are the lightest designs of all. This was not the case for IB and IC. Weights of the ice stringers are between 7,0 and 8,5 % of all ice strengthened structures. Weight differences to designs using the same frame spacing but not including ice stringers are now noticeable but still not significant. Normal frames account 27,0 - 29,5 % of the weight for designs without ice stringers. The design with intermediate ice frames is now the lightest design of all with significantly lighter shell plating.

For longitudinal framing designs the weight continues to increases as the frame spacing decreases. Thicknesses and thus weights of the shell plates are still very similar between the different designs. The frame weight varies between 18,2 % and 28,2 % from the total ice reinforced structure weight. A notable difference from IB and IC classes, is the fact that the largest bulb profile was not enough for certain locations on the hull. T-beam profiles were selected for 2 of the longitudinal designs.

The steel weight of structures outside the scope of study is calculated to be 1227 tonne for ice class IA. By rough estimates for the entire ship, this means that the steel weight increase from the lightest structural design would be 0,72 - 2,88 % between all the other design options.

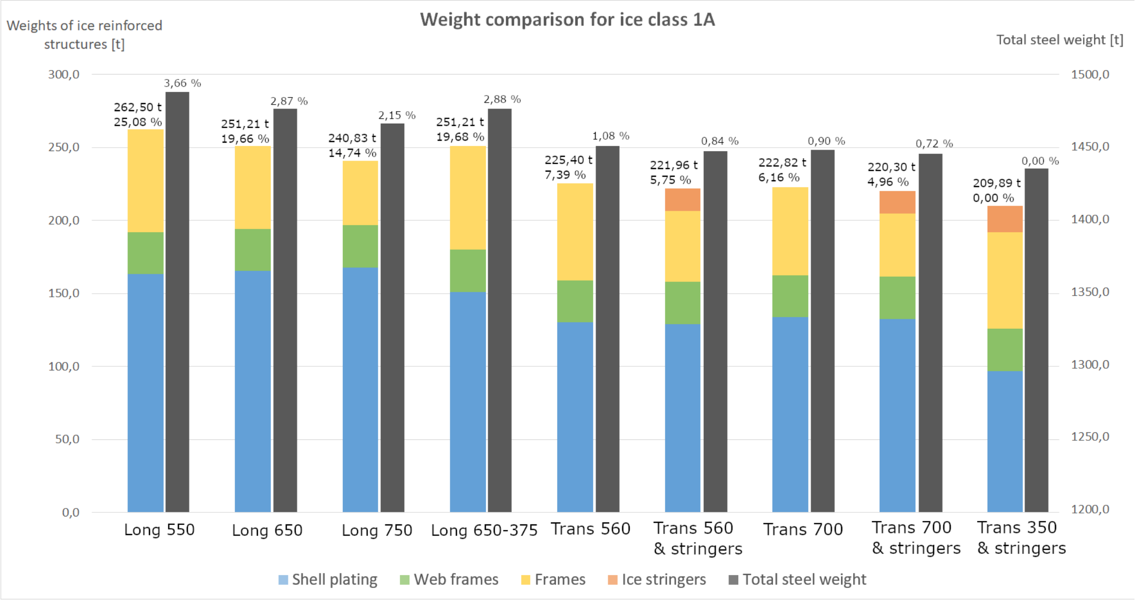


Figure 27: Weight results for ice class IA. Weight increase percentiles from the lightweight option are written above weights of ice reinforced structures and the total steel weight correspondingly.

4.3.4 IA Super

Weight results for ice class IA Super are presented in figure 28. Overall the results have a similar trend in comparison to the results from IA. However the weights increase significantly more. Weights of transverse framing designs increase by 50 - 87 tonne from the corresponding weights of IA class. Weights of longitudinal designs increase by 105 - 130 tonne with a similar comparison.

The lightest structural design by a clear margin is transverse framing with two ice stringers and 350 mm frame spacing. Weight differences to other transversely framed designs are between 18,8 % and 22,6 % which is quite substantial. All of the longitudinal designs are considerably heavier in comparison to any transversely framed options. Heaviest of the designs is longitudinal framing with 550 mm spacing which has a 54,8 % weight difference to the lightest design.

Weight of ice strengthened shell plating is the largest proportion from total weight of ice reinforced structures. The significance of shell plating weight is greatest for ice class IA Super in comparison to the lower ice classes. The shell plating accounts for 54,5 - 71,0 % of the total weight. Weight of the shell plating is the biggest difference between transversely and longitudinally framed designs. Whereas weight of the normal frames is still the biggest difference between designs using the same framing system.

For the first time for transverse framing designs the weight decreases as the frame spacing decreases. Transverse framing designs with ice stringers continue to be the lightest designs of all. Weights of the ice stringers are between 5,7 and 6,7 % of all ice strengthened structures. Weight differences to designs using the same frame spacing but without ice stringers is still not significant. Normal frames account 23,3 - 25,6 % of the weight for designs without ice stringers. The design with intermediate ice frames is now the lightest design by a substantial margin. All transverse framing designs passed the ice class regulations without any T-beam profiles as normal frames.

Variance between the weights of different longitudinal designs decreased as the significance of shell plating increased. The frame weight varies between 19,5 % and 29,0 % from the total ice reinforced structure weight. For the first time the design with 650 - 375 mm spacing became the lightest longitudinal design. None of the longitudinally framed designs passed the ice class regulations without T-beam profiles as normal frames.

The steel weight of structures outside the scope of study is calculated to be 1193 tonne for ice class IA Super. By rough estimates for the entire ship, this means that the steel weight increase from the lightest structural design would be 3,29 - 9,56 % between all the other design options.

Figure 28: Weight results for ice class IA Super. Weight increase percentiles from the lightweight option are written above weights of ice reinforced structures and the total steel weight correspondingly.

4.3.5 Validation of results

To validate the results and to confirm that the design and optimization tool works as intended, couple cross-checking procedures were performed. First check was to ensure that the output files written by the CAESES script were filled with correct and accurate information. One longitudinal and one transverse design output were double checked entirely. The dimensions were checked to be moderate and the location information of each individual part was checked to be correct. Because the script also displays the designs on top of the hull form, all of the designs could be checked over to ensure that they were created correctly.

Second check was performed for the Python code. All of the calculation functions were cross-checked with separate calculations. This was done to ensure that each function worked as intended and followed the ice class rules properly.

Third and final check was done by comparing the calculated weight results with the weight data from the reference steel model. Weights of framing and plating structures that were stripped from the steel model did match with the weights of ice strengthened structures selected by the tool for lowest ice class. Because these weights were closely matched, it can be argued that the tool works properly and that the results are indeed valid. The differences between the steel model weight and calculated weight can be explained by different shell plating and stiffener profiling.

5 Conclusion and discussion

The purpose of this thesis work was to study the effects of structural design selections on steel weight of ship's ice strengthened hull structures. The research was conducted as a case study using a concept design vessel. A design and optimization tool was developed to obtain weight results for different framing designs. Ship particulars were taken from the concept design's project material and dimensions of different structures were measured using the case study ship's 3D hull form model. Scantling requirements were determined according to Finnish-Swedish Ice Class Rules. Scantling selection and weight calculations were done using a pre-defined library of steel profiles and -plates.

Weight differences between all the studied structural designs were relatively minor for ice classes IC and IB. The chosen framing system had the greatest impact on weight of all the other design selections. It can be noted that transverse framing systems were systematically lighter in comparison to longitudinal framing systems. Other design selections had no significant effects on weight. A trend of increasing weights with decreasing frame spacing and with added ice stringers could also be noted for both framing systems. This is because the biggest factor in terms of steel weight was the thickness of shell plating. The selected shell thicknesses were quite similar despite the varying frame spacing. In addition all the selected frames were mostly the lightest profiles available. This explains why the added frames and ice stringers only increased weight. It can concluded that the weight differences of ice strengthened hull structures are relatively insignificant for ice classes IC and IB. It can be further argued that the only design selection worth considering is the framing system if the ship in question has a lower ice class.

Weight differences started to increase for ice class IA and especially for ice class IA Super. The importance of framing system further increased as transverse framing systems became notably lighter in comparison to longitudinal framing systems. The effects of other design selections also started to appear as ice classes were higher. For transversely framed designs the weight decreased as the framing structure complexity increased with lower frame spacing and with added ice stringers. Lower frame spacing had greater impact on weight in comparison to added ice stringers. The reason for this is the previously mentioned shell plate thickness which accounts for most of the steel weight. Lower frame spacing allowed the plating thickness to be slightly lower. Ice stringers reduced weight as they allowed frames to terminate at more desirable locations and thus allowed lighter bulb profiles to be used as frames. For longitudinally framed designs the weights did not decrease in a similar manner with decreasing frame spacing. Plate thicknesses were really high for all longitudinally framed designs. Even with the lowest frame spacing of 375 mm, the plate thickness could only be brought to the same thickness level as was used for the largest transverse frame spacing. Additional problem with the longitudinal framing systems was the fact that there were no bulb profiles available which could fulfil the ice class rules. T-beam profiles were selected on certain locations for two IA class designs and all four IA Super designs with longitudinal framing. The problem is not the T-beam profiles themselves but the space that is required for their welding. In tight spaces and with low frame spacing, their welding can become impossible. It can be concluded that the weight differences become at least moderately significant for ice classes IA and IA Super. The effects of ice strengthening should definitely be considered when the framing system of a ship is decided upon. Other design selections on the framing structures can also be considered with the effects of ice strengthening in mind. Intermediate ice frames and well placed ice stringers can save weight especially for ice class IA Super.

Further discussion on differences between the two framing systems can also be brought up. As mentioned before, transversely framed designs were systematically always lighter in comparison to longitudinally framed designs. One reason for this is the fact that ice class rules which have tougher strength requirements for longitudinal framing systems. This is because individual structures within longitudinal designs are subjected to greater stresses from ice actions as the structure members are parallel to the ice sheets and do not distribute the ice loads as well as transverse designs. Second reason is the frame spacing which is used for scantling calculations. The spacing between frames grow in length especially for longitudinal frames in the bow section as the hull surface deforms. The growth of spacing was not that severe between transverse frames. It can be argued that well placed longitudinal ice frames could have made a notable difference to the weights of longitudinal designs.

It is also important to acknowledge the different assumptions and simplifications that influenced the results. First of all it has to be noted that the case study was conducted with only one hull form and only nine different structural designs. For this reason the results are fairly accurate for this particular ship but only provide general information of how ice strengthened hull structures could affect the steel weight of other ships. Second notable simplification is the fact that different structural designs were only fitted between the bottom and the second deck. The remaining steel weight information was taken from the preliminary steel model of the case study vessel. Because of this the results should only be taken as rough estimates for the entire ship. Last but not least, the only set of rules applied to the studied structures were the Finnish-Swedish Ice class rules. Thus the scantling selection for frame parts located outside of the ice reinforced region is done using the ice class rules. Although the proportion of frame parts affected by this is relatively small, it still downplays the positive effects of ice stringers for example.

Over all it can concluded that this type of optimization and analysis can become a valuable part of parametric ship design process. The developed design and optimization tool fits well with needs of parametric design. Flexible and fast analysis of any hull shape is definitely a desired feature for a parametric design tool. This way valuable information can be extracted from a number of different options with relative ease. Naturally the current version of the design and optimization tool is far from optimal. The biggest improvement for future development of the tool would be to include other classification society rules and expand the research to account more ship structures. This way a more holistic view on the effects of different design options could be achieved. Alternatively the tool could be improved by including smaller structures such as brackets, and smaller details such as angles between stiffening profiles and plates, into the analysis.

In terms of the research work, future studies could also be developed further. Similar improvements as mentioned for the design and optimization tool could be implemented into the study as well. This would increase the accuracy and complexity of the results. Another way to improve the current research work could be accomplished by including more case study hull forms and by increasing the number of structural designs. This way the research could be expanded from one case study ship to a whole fleet of different ship types.

Bibliography

Alanko, J. 2007. *Laivan yleissuunnittelu.* Turku, Finland: Karhukopio.

Babichz, J. 2015. *Wärtsilä Encyclopedia of Ship Technology, Second Edition.* Wärtsilä Corporation. ISBN: 978-952-93-5536-5.

Branke, J., Deb, S., Miettinen, K., Roman, S., 2008. *Multiobjective Optimization, Interactive and Evolutionary Approaches.* Springer Germany. doi: 10.1007/978-3-540-88908-3. ISBN: 3-540-88907-8.

Continental Steel Pte Ltd, 2000. *Products Handbook, Structural Steel.* Available at: https://mafiadoc.com/products-handbook-continental-steel-pte-ltd_59d22b0f1723dd85e13a35dd.html (Accessed: 13 April 2020).

Daley, C., Riska, K., Tuhkuri, J. 1998. 'The role of discrete failures in local ice loads', *Cold Regions Science and Technology.* Volume 27, Pages 197-211. doi: 10.1016/S0165-232X(98)0007-X.

Frederking, R., 1999. 'The Local Pressure-Area Relation in Ship Impact with Ice', in the *15th International Conference on Port and Ocean Engineering under Arctic Conditions, POAC'99.* Volume 2, Pages 687-696, Helsinki, Finland.

Friendship Systems AG, 2020. *CAESES product overview.* Available at: https://www.caeses.com/products/caeses/ (Accessed: 8 May 2020).

Jordaan, I. 2001. 'Mechanics of ice-structure interactions', *Engineering Fracture Mechanics.* Volume 68, Issue 17-18, Pages 1923-1960. doi:10.1016/S0013-7944(01)00032-7. ISSN: 00137944.

Kishk, A., Mikki, S. 2008. *Particle Swarm Optimization: A Physics-Based Approach.* Morgan Claypool. Pages 1-7 doi: 10.2200/S00110ED1V01Y200804CEM020. ISBN: 9781598296143.

Kujala, P., 1994 'On the statistics of ice loads on ship hull in the Baltic', Dissertation. *Mechanical Engineering Series, Number 116.* Page 98.

Lewis, E.V., 1988. *Principles of Naval Architecture, Second Revision, Volume I, Section 2.* Society of Naval Architects and Marine Engineers. ISBN: 978-0-939773-00-8.

Lloyd’s Register Group, 2020. *SOLAS - International Convention for the Safety of Life at Sea.* Available at: https://www.lr.org/en/rulefinder (Accessed 1 May 2020).

Mandal, N.R., 2017. *Ship Construction and Welding, Springer Series on Naval Architecture, Volume 2.* Pages 25-38. Springer. doi: 10.1007/978-981-10-2955-4. ISBN: 978-981-10-2955-4

Papanikolaou, A., 2010. ’Holistic Ship design optimization’, *CAD Computer Aided Design.* Volume 42, Issue 11, Pages 1028-1044. doi: 10.1016/j.cad.2009.07.002. ISSN: 00104485.

Python Software Foundation, 2020. *General Python FAQ.*Available at: https://docs.python.org/3/faq/general.html (Accessed: 2 May 2020).

Rautaruukki Oyj, 2006. *Hot Rolled Shipbuilding profiles.* Available at: https://pdfslide.net/documents/shipbuilding-profiles-en-0806.html (Accessed: 11 April 2020).

Riska, K., Kujala, P., 2010. *Talvimerenkulku.* Helsinki University of Technology. ISBN: 9789526030166.

Riska, K., 2019. *Ship-Ice Interaction in Ship Design: Theory and Practice.* Available at: https://pdfs.semanticscholar.org/c14d/add4eb6da9e68e5f4be4ff6c18acbd90a742.pdf

Rodríguez, A., Fernández-Jambrina, L., 2012. ’Programmed design of ship forms’, *CAD Computer Aided Design.* Volume 44, Issue 7, Pages 687-696. doi: 10.1016/j.cad.2012.03.003. ISSN: 00104485.

Sanderson, T.J., 1988. *Ice Mechanics and Risks to Offshore Structures, first edition.* Springer Netherlands. ISBN: 978-0-86010-785-9.

Traficom, 2019. *Guidelines for the application of the 2017 Finnish-Swedish Ice Class Rules.* Finnish Transport and Communications Agency & Swedish Transport Agency. TRAFI/708629/03.04.01.01/2018. Available at: https://www.traficom.fi/sites/default/files/media/file/FSICR%20Guidelines%202018.pdf (Accessed: 5. April 2020).

Traficom, 2017. *Ice Class Regulations and the Application Thereof.* Finnish Transport and Communications Agency & Swedish Transport Agency. TRAFI/494131/03.04.01.00/2016. Available at: https://www.traficom.fi/en/transport/maritime/ice-classes-ships (Accessed: 5. April 2020).

A Finnish-Swedish Ice Class Rules used in the thesis work

A.1 Definitions

L	m	the length of the ship between the perpendiculars
L_{Bow}	m	the length of the bow
L_{Par}	m	the length of the parallel midship body
B	m	the maximum breadth of the ship
T	m	the actual ice class draughts of the ship (UIWL)
A_{wf}	m^2	the area of the waterline of the bow
α	degree	the angle of the waterline at B/4
ϕ_1	degree	the rake of the stem at the centerline
ϕ_2	degree	the rake of the bow at B/4
ψ	degree	the flare angle calculated as $\psi = \tan^{-1}\frac{\tan(\sigma)}{\sin(\alpha)}$ using local angles α and σ at each location. $\sigma = \sigma_2$
D_P	m	the diameter of the propeller
H_m	m	the thickness of the brash ice in mid channel
H_F	m	the thickness of the brash ice layer displaced by the bow

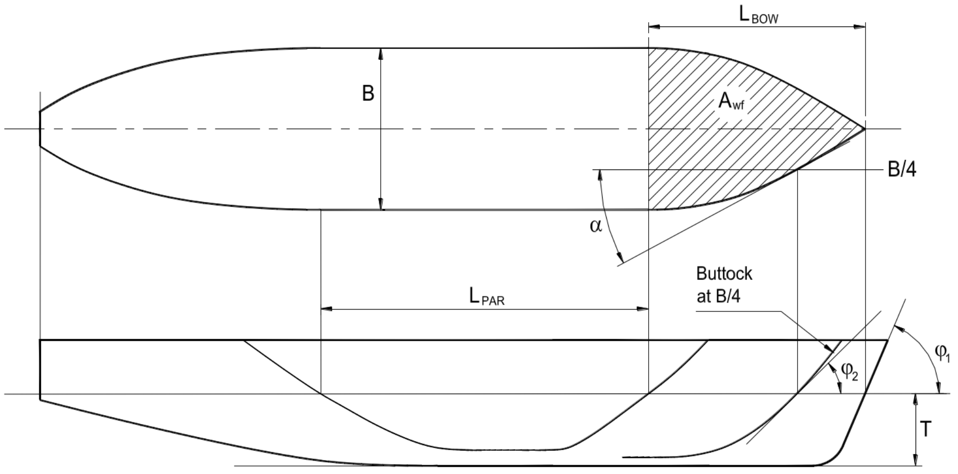


Figure A1: Geometric hull form definitions.

A.2 Engine output

$$P_{min} = K_e \frac{(R_{CH}/1000)^{3/2}}{D_p} \qquad [\text{kW}] \tag{A1}$$

K$_e$ shall be given a value according to table A1.

K_e values		
Number of propellers	CP propeller or electric or hydraulic propulsion machinery	FP propeller
1	2,03	2,26
2	1,44	1,60
3	1,18	1,31

Table A1: Values of K$_e$ for conventional propulsion systems

$$R_{CH} = C_1 + C_2 + C_3 C_\mu (H_F + H_M)^2 (B + C_\psi H_F) + C_4 L_{Par} H_F^2 + C_5 (\frac{LT}{B^2})^3 \frac{A_{wf}}{L} \tag{A2}$$

where,

C_μ = $0.15\cos(\phi_2) + \sin(\psi)\sin(\alpha)$, $C_\mu \geq 0.45$
C_ψ = $0.047\psi - 2.115$ and $C_\psi = 0$ if $\psi \geq 45°$
H_F = $0.26 + (H_M B)^{0.5}$
H_M = 1.0 m for ice classes IA and IA Super
= 0.8 m for ice class IB
= 0.6 m for ice class IC

C_1 = 0 for ice classes IA, IB, and IC
C_2 = 0 for ice classes IA, IB, and IC

For ice class IA Super:

$$C_1 = f_1 \frac{BL_{Par}}{2\frac{T}{B}+1} + (1 + 0.021\phi_1)(f_2 B + f_3 L_{Bow} + f_4 B L_{Bow} \tag{A3}$$

$$C_2 = (1 + 0.063\phi_1)(g_1 + g_2 B) + g_3 (1 + 1.2\frac{T}{B})\frac{B^2}{\sqrt{L}} \tag{A4}$$

C_3 = 845 [kg/(m^{2}s^2)]
C_4 = 42 [kg/(m^{2}s^2)]
C_5 = 825 [kg/s^2]
ψ = $\tan^{-1}(\frac{\tan(\phi_2}{\sin(\alpha)})$

$5 \leq {\frac{\mathrm{LT}}{\mathrm{B}^2}}^3 \leq 20$

Coefficients f_1-f_4 and g_1-g_3 are given in table A2.

f-coefficients	
f_1	23 [N/m^2]
f_2	45,8 [N/m]
f_3	14,7 [N/m]
f_4	29 [N/m^2]

g-coefficients	
g_1	1530 [N]
g_2	170 [N/m]
g_3	400[N/m^1.5]

Table A2: Values of coefficients f_1-f_4 and g_1-g_3 for determination C_1 and C_2.

A.3 Ice load

Heights of the ice load areas are listed in table A3.

Ice Class	**h$_i$**	**h**
#	m	m
IA Super	1,0	0,35
IA	0,8	0,30
IB	0,6	0,25
IC	0,4	0,22

Table A3: Values of h_i and h for the different ice classes.

The design ice pressure is determined by formula:

$$p = c_d c_p c_a p_0 \qquad [\mathrm{MPa}] \tag{A5}$$

where,

$$c_d = \frac{ak+b}{1000}, \qquad c_d \leq 1.0 \tag{A6}$$

where,

$$k = \frac{\sqrt{\Delta P}}{1000} \tag{A7}$$

Values of a and b are listed in table A4

Δ	is the displacement of the ship at a maximum ice class draught [t]
P	is the actual continuous engine output [kW]
c_p	value is given in table A5.

	Bow		**Midbody and stern**	
	k $\leq$ 12	k >12	k $\leq$ 12	k >12
a	30	6	8	2
b	230	518	214	286

Table A4: Values of a and b for the different hull regions.

	Bow	**Midbody**	**Stern**
IA Super	1,0	1,00	0,75
IA	1,0	0,85	0,65
IB	1,0	0,70	0,45
IC	1,0	0,50	0,25

Table A5: Values of c_p for the different hull regions.

$$c_a = \sqrt{\frac{l_0}{l_a}}, \qquad 0.35 \leq c_a \geq 1.0 \tag{A8}$$

$l_0 = 0.6$
$p_o = 5.6$ [MPa]
l_a value is given in table A6

Structure	**Framing system**	**l_a [m]**
Shell	Transverse	Frame spacing
	Longitudinal	1.7 x Frame spacing
Frames	Transverse	Frame spacing
	Longitudinal	Span of frame
Ice stringer	-	Span of stringer
Web frame	-	2 x Web frame spacing

Table A6: Values of l_a for different structural elements.

A.4 Hull regions

Ship's hull is divided into three regions as follows (illustrated in figure A2):

Bow region: From the stem to a line parallel to and 0.04L aft of the forward borderline of the part of the hull where the waterlines run parallel to the centerline. For ice classes IA Super and IA, the overlap over the borderline need not exceed 6 metres, for ice classes IB and IC this overlap need not exceed 5 metres.

Midbody region: From the aft boundary of the Bow region to a line parallel to and 0.04L aft of the aft borderline of the part of the hull where the waterlines run parallel to the centerline. For ice classes IA Super and IA, the overlap over the borderline need not exceed 6 metres, for ice classes IB and IC this overlap need not exceed 5 metres.

Stern region: From the aft boundary of the Midbody region to the stern.

L shall be taken as the ship's rule length used by the classification society.

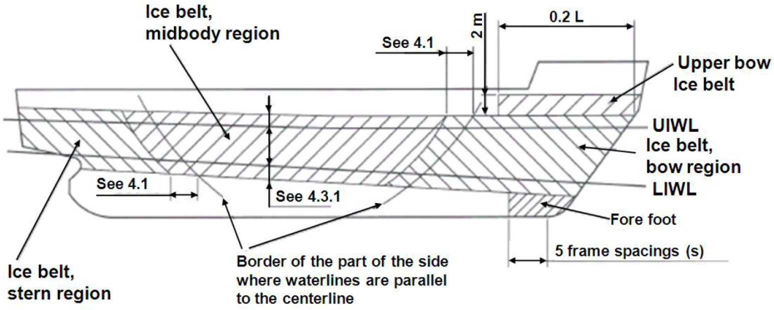


Figure A2: Ice strengthened regions of the hull

Vertical extensions for ice strengthened structures are given in tables A7 and A8.

Ice Class	Hull region	Above UIWL	Below LIWL
#	#	m	m
256 height IA Super	Bow	0.60	1.20
	Midbody		
	Stern		1.0
IA	Bow	0.50	0.90
	Midbody		0.75
	Stern		
IB and IC	Bow	0.40	0.70
	Midbody		0.60
	Stern		

Table A7: Vertical extensions of ice strengthened shell plates (ice belt).

Ice Class	Hull region	Above UIWL	Below LIWL
#	#	m	m
IA Super	Bow	1.2	Down to tank top or below top of the floor
	Midbody		2.0
	Stern		1.6
IA, IB, and IC	Bow	1.0	1.6
	Midbody		1.3
	Stern		1.0

Table A8: Vertical extensions of ice strengthened framing structures.

A.5 Scantling requirements

Plate thickness in the ice belt

For transverse framing, the thickness of the shell plating shall be determined by the formula:

$$t = 667s\sqrt{\frac{f_1 p_{pl}}{\sigma_y}} + t_c, \qquad \text{[mm]} \tag{A9}$$

and for longitudinal framing, the thickness of the shell plating shall be determined by the formula:

$$t = 667s\sqrt{\frac{p}{f_2 \sigma_y}} + t_c, \qquad \text{[mm]} \tag{A10}$$

where,

s is the frame spacing [m]
p_{pl} = 0.75p [MPa], where p is ice pressure
f_1 = 1.3 - $\frac{4.2}{\mathrm{h/s+1.8^2}}$ and $\mathrm{f}_1 \leq 1.0$

f_2 = 0.6 + $\frac{0.4}{\mathrm{h/s}}$ when $\mathrm{h/s} \leq 1.0$
= 1.4 - 0.4(h/s) when $1 \leq \mathrm{h/s} \leq 1.8$

h is the design ice load height [m]
σ_y = 315 [N/mm^2] for high strength structural steel
t_c = 2 [mm]

Transverse frames

The section modulus of a main or intermediate transverse frame shall be calculated using the formula:

$$Z = \frac{pshl}{m_t \sigma_y} \times 10^6 \qquad \text{[cm}^3\text{]} \tag{A11}$$

and the effective shear area will be calculated from:

$$A = \frac{\sqrt{3} f_3 phs}{2\sigma_y} \times 10^4 \qquad \text{[cm}^2\text{]} \tag{A12}$$

where,

p is the ice pressure [MPa]
s is the frame spacing [m]
h is the design ice load height [m]
l is the frame span [m]
σ_y is the yield stress [N/mm^2]
$m_t = \frac{7m_0}{7-5h/l}$
$f_3 = 1.2$
$m_0 = 7$ (boundary condition)

Longitudinal frames

The section modulus of longitudinal frame shall be calculated using the formula:

$$Z = \frac{f_4 p h l^2}{m \sigma_y} \times 10^6 \qquad [\mathrm{cm}^3] \tag{A13}$$

and the effective shear area is be calculated from:

$$A = \frac{\sqrt{3} f_4 f_5 p h l}{2 \sigma_y} \times 10^4 \qquad [\mathrm{cm}^2] \tag{A14}$$

where,

$f_4 = (1 - 0.2\ h/s)$
$f_5 = 2.16$
$m = 13.3$
p is the ice pressure [MPa]
s is the frame spacing [m]
h is the design ice load height [m]
l is the frame span [m]
σ_y is the yield stress [N/mm^2]

Ice stringers within the ice belt

The section modulus of a stringer situated within the ice belt shall be calculated using the formula:

$$Z = \frac{f_6 f_7 p h l^2}{m \sigma_y} \times 10^6 \qquad [\mathrm{cm}^3] \tag{A15}$$

and the effective shear area is calculated from:

$$A = \frac{\sqrt{3} f_6 f_7 f_8 p h l}{2 \sigma_y} \times 10^4 \qquad [\mathrm{cm}^2] \tag{A16}$$

where,

p is the ice pressure [MPa]
h is the design ice load height [m]
$ph \geq 0.15$ [MN/m]
l is the stringer span [m]
σ_y is the yield stress [N/mm^2]
$f_6 = 0.9$
$f_7 = 1.8$
$f_8 = 1.2$
$m = 13.3$

Ice stringers outside the ice belt

The section modulus of a stringer situated outside the ice belt but supporting ice-strengthened frames shall be calculated using the formula:

$$Z = \frac{f_9 f_{10} p h l^2}{m \sigma_y} \times (1 - h_s/l_s) \times 10^6 \qquad [\mathrm{cm}^3] \tag{A17}$$

and the effective shear area is calculated from:

$$A = \frac{\sqrt{3} f_9 f_{10} f_{11} p h l}{2 \sigma_y} \times (1 - h_s/l_s) \times 10^4 \qquad [\mathrm{cm}^2] \tag{A18}$$

where,

p is the ice pressure [MPa]
h is the design ice load height [m]
$ph \geq 0.15$ [MN/m]
l is the stringer span [m]
l_s is the distance to adjacent ice stringer or deck [m]
h_s is the distance to the ice belt [m]
σ_y is the yield stress [N/mm^2]
$f_9 = 0.80$
$f_{10} = 1.8$
$f_{11} = 1.2$
$m = 13.3$

Web frames

The ice load transferred to a web frame from an ice stringer or from longitudinal framing shall be calculated using the formula:

$$F = f_{12} p h S \qquad [\mathrm{MN}] \tag{A19}$$

where,

p is the ice pressure [MPa]
h is the design ice load height [m]
ph $\geq$ 0.15 [MN/m]
S is the distance between web frames [m]
f_{12} = 1.8

Effective shear area is calculated from:

$$A = \frac{\sqrt{3}\alpha f_{13} Q}{\sigma_y} \times 10^4 \qquad [\mathrm{cm}^2] \tag{A20}$$

where,

Q is the maximum calculated shear force undrer the ice load F
σ_y is the yield stress [N/mm^2]
α is given in table A9
f_{13} = 1.1

Section modulus:

$$Z = \frac{M}{\sigma_y}\sqrt{\frac{1}{1-(\gamma A/A_a)^2}} \times 10^6 \qquad [\mathrm{cm}^3] \tag{A21}$$

where,

M = 0.193Fl
γ is given in table A9
A is the required shear area
A_a is the actual cross sectional area of the web frame, $A_a = A_f + A_w$
α is given in table A9

A_f/A_w	**0.0**	**0.2**	**0.4**	**0.6**	**0.8**	**1.0**	**1.2**	**1.4**	**1.6**	**1.8**	**2.0**
α	1.5	1.23	1.16	1.11	1.09	1.07	1.06	1.05	1.05	1.04	1.04
γ	0	0.44	0.62	0.71	0.76	0.80	0.83	0.85	0.87	0.88	0.89

Table A9: Values of factors α and γ

where,

A_f is the actual cross-sectional area of the free flange
A_w is the actual effective cross-sectional area of the web plate

B Section modulus calculations

Elastic section modulus calculation is performed for all T-beam and bulb profiles using the formula:

$$Z_e = \frac{I}{z} \tag{B1}$$

where,

I	is the combined second moment of inertia for the stiffening profile and the shell plate
z	is the distance from the combined neutral axis to the furthest point

Figure B1 illustrates how different variables are defined for combined second moment of inertia and combined neutral axis calculations. Second moment of inertia, cross sectional area, and neutral axis locations are pre-defined for all the steel profiles.

Combined second moment of inertia is calculated using the formula:

$$I = (I_{profile} + A_{profile} h_{profile}^2) + (I_{plate} + A_{plate} h_{plate}^2) \tag{B2}$$

and combined neutral axis is calculated using the formula:

$$y_n = (A_{profile} \times y_{profile} + A_{plate} \times y_{plate})/(A_{profile} + A_{plate}) \tag{B3}$$

where,

$I_{profile}$	is the profile's second moment of inertia
$A_{profile}$	is the profile's cross sectional area
$y_{profile}$	is the profile's neutral axis location plus plate thickness t
t	is the plate thickness
b	is the effective breadth of shell plating
I_{plate}	is the plate's effective second moment of inertia, $I_{plate} = \frac{bt^3}{12}$
A_{plate}	is the plate's effective cross sectional area, $A_{plate} = bt$
y_{plate}	is the plate's neutral axis location, $y_{plate} = \frac{t}{2}$
h_{plate}	is the distance between the plate's neutral axis and the combined neutral axis, $h_{plate} = y_n - y_{plate}$
$h_{profile}$	is the distance between the profile's neutral axis and the combined neutral axis, $h_{profile} = y_{profile} - y_n$

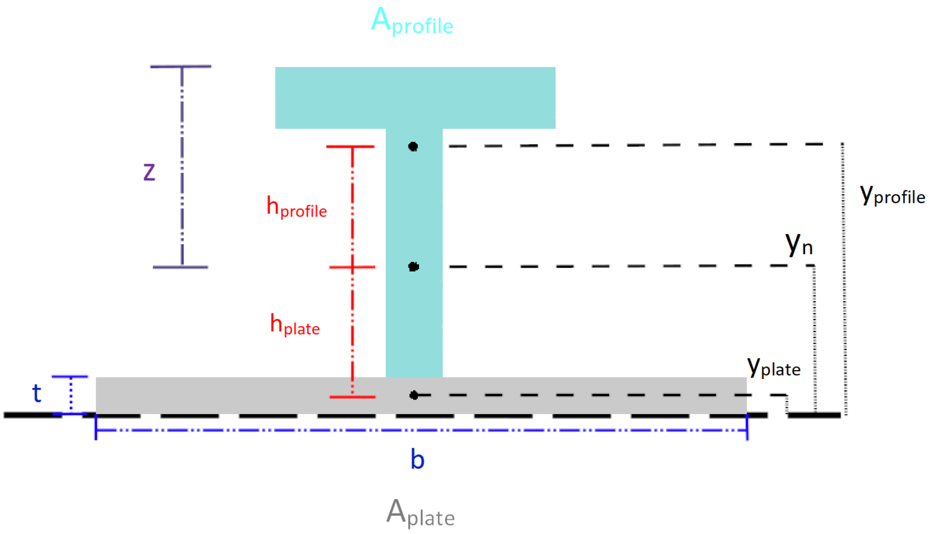


Figure B1: Definitions for combined second moment of inertia and combined neutral axis calculations.

C Steel plate and profile libraries

Bulp profiles									
ID	**a**	**s**	**c**	**d**	**r**	**Area**	**Weight**	**Cx**	**Ix**
#	mm	mm	mm	mm	mm	cm2	kg/m	mm	cm4
HP140x10	140	10	19,0	19,7	5,5	16,63	13,05	7	5,56
HP160x9	160	9	22,0	22,2	6	17,8	13,97	7,1	7,32
HP180x8	180	8	25,0	25,5	7	18,86	14,8	7,4	9,9
HP180x9	180	9	25,0	25,5	7	20,66	16,22	7,7	10,93
HP180x10	180	10	25,0	25,5	7	22,46	17,63	8,1	12,05
HP200x9	200	9	28,0	28,8	8	23,66	18,57	8,4	15,76
HP200x10	200	10	28,0	28,8	8	25,66	20,14	8,7	17,21
HP220x10	220	10	31,0	32,1	9	29	22,77	9,3	23,89
HP200x12	200	12	28,0	28,8	8	29,66	23,28	9,4	20,46
HP240x10	240	10	34,0	35,4	10	32,49	25,5	10	32,34
HP220x12	220	12	31,0	32,1	9	33,4	26,22	10	27,98
HP240x11	240	11	34,0	35,4	10	34,89	27,39	10,3	34,81
HP260x10	260	10	37,0	38,7	11	36,11	28,35	10,7	42,84
HP240x12	240	12	34,0	35,4	10	37,29	29,27	10,6	37,43
HP260x11	260	11	37,0	38,7	11	38,71	30,39	11	45,9
HP260x12	260	12	37,0	38,7	11	41,31	32,43	11,3	49,11
HP280x11	280	11	40,0	42	12	42,68	33,5	11,7	59,44
HP280x12	280	12	40,0	42	12	45,48	35,7	11,9	63,34
HP300x11	300	11	43,0	45,3	13	46,78	36,7	12,4	75,74
HP300x12	300	12	43,0	45,3	13	49,79	39,09	12,6	80,44
HP300x13	300	13	43,0	45,3	13	52,79	41,44	12,9	85,33
HP320x12	320	12	46,0	48,6	14	54,25	42,6	13,4	100,8
HP320x13	320	13	46,0	48,6	14	57,45	45,09	13,6	106,6
HP340x12	340	12	49,0	52	15	58,84	46,2	14,1	124,6
HP340x14	340	14	49,0	52	15	65,54	51,5	14,6	138,6
HP370x13	370	13	53,5	56,9	16,5	69,7	54,7	15,4	176,7
HP370x15	370	15	35,5	56,9	16,5	77,1	60,5	15,9	194,8
HP400x14	400	14	58,0	61,9	18	81,48	63,96	16,8	243,6
HP400x16	400	16	58,0	61,9	18	89,48	70,2	17,2	266,6
HP430x14	430	14	62,5	66,8	19,5	89,7	70,6	17,9	313,9
HP430x15	430	15	62,5	66,8	19,5	94,19	73,9	18,1	327,9
HP430x17	430	17	62,5	66,8	19,5	102,79	80,7	18,5	356,7

Table C1: Bulp profile library

T-beam profiles									
ID	**B**	**d**	**t**	**T**	**r**	**Area**	**Weight**	**Cx**	**Ix**
#	mm	mm	mm	mm	mm	cm2	kg/m	mm	cm4
200x140	142,20	201,50	6,80	11,20	10,20	29,30	23,00	151,30	1117,00
175x175	175,00	175,00	7,00	11,00	13,00	31,50	24,70	137,40	811,00
200x175	177,70	201,20	7,70	10,90	10,20	34,50	27,10	152,90	1289,00
230x155	154,40	230,90	9,60	17,00	10,20	47,20	37,10	172,10	2325,00
225x200	200,00	225,00	9,00	14,00	18,00	48,40	38,00	173,50	2144,00
265x165	166,50	267,50	10,29	16,51	12,70	54,00	42,41	195,10	3745,00
230x190	191,90	231,60	10,50	17,70	10,20	56,90	44,60	176,90	2679,00
250x200	200,00	250,00	10,00	16,00	20,00	57,10	44,80	190,40	3190,00
300x175	178,80	301,50	10,92	14,99	12,70	58,80	46,13	213,60	5427,00
300x200	200,00	300,00	11,00	17,00	22,00	67,20	52,80	221,60	5749,00
270x210	210,80	269,70	11,60	18,80	12,70	69,40	54,50	203,60	4591,00
300x230	230,20	308,50	13,10	22,10	12,70	89,10	69,90	232,40	7724,00
340x250	253,70	341,90	12,45	18,92	15,20	89,20	69,94	255,20	9906,00
300x300	300,00	294,00	12,00	20,00	28,00	96,20	75,60	233,20	6679,00
350x250	255,80	346,60	14,48	23,62	15,20	108,00	84,83	259,70	12010,00
380x265	266,60	381,20	14,35	21,59	16,50	110,00	86,31	281,20	15470,00
350x300	300,00	350,00	13,00	24,00	28,00	118,00	92,40	274,50	11970,00
420x290	292,40	420,20	14,73	21,72	17,80	124,00	96,70	309,20	21260,00
400x300	300,00	400,00	14,00	26,00	28,00	134,00	105,00	308,20	18690,00
350x350	355,00	348,00	15,40	24,80	15,00	139,00	109,00	273,10	14010,00
450x300	300,00	450,00	16,00	28,00	18,00	153,00	120,00	336,00	29030,00
390x380	382,00	390,00	18,00	30,10	17,00	181,00	142,00	305,00	22840,00
500x300	300,00	500,00	19,10	35,90	29,00	200,00	157,00	371,00	46440,00
430x400	401,00	431,00	19,70	32,40	18,00	210,00	165,00	334,00	33240,00
500x400	400,00	500,00	19,10	36,10	29,00	237,00	186,00	388,00	50380,00
460x420	420,00	461,00	21,30	36,60	24,00	247,00	194,00	358,00	43990,00
500x450	451,00	500,00	19,10	35,90	29,00	254,00	199,00	395,00	51900,00

Table C2: T-beam profile library

shell plate thicknesses
t
mm
10
12
14
16
18
20
22
25
28
30
32
35
38
40
42
45
48
50

Table C3: Shell plate thickness library

D Weight results in table format

Long 550										
Ice reinforcements								Estimates for the entire ship		
Plates [kg]	Frames [kg]	Ice Stringers [kg]	Web Frames [kg]	Total [kg & t]		Weight difference to the lightweight option [t]	Weight increase from the lightweight option (only ice reinforcements considered)	Estimated total steel weight [t]	Weight of ice reinforcements divided by estimated total steel weight	Weight increase from the lightweight option
118908	49195	0	22689	190793	190,79	20,40	11,97 %	1435,79	13,29 %	1,44 %
62,32 %	25,78 %	0,00 %	11,89 %							

Long 650										
Ice reinforcements								Estimates for the entire ship		
Plates [kg]	Frames [kg]	Ice Stringers [kg]	Web Frames [kg]	Total [kg & t]		Weight difference to the lightweight option [t]	Weight increase from the lightweight option (only ice reinforcements considered)	Estimated total steel weight [t]	Weight of ice reinforcements divided by estimated total steel weight	Weight increase from the lightweight option
120773	41339	0	22744	184856	184,86	14,46	8,49 %	1429,86	12,93 %	1,02 %
65,33 %	22,36 %	0,00 %	12,30 %							

Long 750										
Ice reinforcements								Estimates for the entire ship		
Plates [kg]	Frames [kg]	Ice Stringers [kg]	Web Frames [kg]	Total [kg & t]		Weight difference to the lightweight option [t]	Weight increase from the lightweight option (only ice reinforcements considered)	Estimated total steel weight [t]	Weight of ice reinforcements divided by estimated total steel weight	Weight increase from the lightweight option
123589	34111	0	22744	180444	180,44	10,05	5,90 %	1425,44	12,66 %	0,71 %
68,49 %	18,90 %	0,00 %	12,60 %							

Long 650 - 375										
Ice reinforcements								Estimates for the entire ship		
Plates [kg]	Frames [kg]	Ice Stringers [kg]	Web Frames [kg]	Total [kg & t]		Weight difference to the lightweight option [t]	Weight increase from the lightweight option (only ice reinforcements considered)	Estimated total steel weight [t]	Weight of ice reinforcements divided by estimated total steel weight	Weight increase from the lightweight option
113591	55142	0	22656	191389	191,39	21,00	12,32 %	1436,39	13,32 %	1,48 %
59,35 %	28,81 %	0,00 %	11,84 %							

Trans 560										
Ice reinforcements								Estimates for the entire ship		
Plates [kg]	Frames [kg]	Ice Stringers [kg]	Web Frames [kg]	Total [kg & t]		Weight difference to the lightweight option [t]	Weight increase from the lightweight option (only ice reinforcements considered)	Estimated total steel weight [t]	Weight of ice reinforcements divided by estimated total steel weight	Weight increase from the lightweight option
95315	55575	0	22593	173483	173,48	3,09	1,81 %	1418,48	12,23 %	0,22 %
54,94 %	32,03 %	0,00 %	13,02 %							

Trans 560 & ice stringers										
Ice reinforcements								Estimates for the entire ship		
Plates [kg]	Frames [kg]	Ice Stringers [kg]	Web Frames [kg]	Total [kg & t]		Weight difference to the lightweight option [t]	Weight increase from the lightweight option (only ice reinforcements considered)	Estimated total steel weight [t]	Weight of ice reinforcements divided by estimated total steel weight	Weight increase from the lightweight option
95315	41772	14974	22593	174654	174,65	4,26	2,50 %	1419,65	12,30 %	0,30 %
54,57 %	23,92 %	8,57 %	12,94 %							

Trans 700										
Ice reinforcements								Estimates for the entire ship		
Plates [kg]	Frames [kg]	Ice Stringers [kg]	Web Frames [kg]	Total [kg & t]		Weight difference to the lightweight option [t]	Weight increase from the lightweight option (only ice reinforcements considered)	Estimated total steel weight [t]	Weight of ice reinforcements divided by estimated total steel weight	Weight increase from the lightweight option
98468	49331	0	22593	170392	170,39	0,00	0,00 %	1415,39	12,04 %	0,00 %
57,79 %	28,95 %	0,00 %	13,26 %							

Trans 700 & ice stringers										
Ice reinforcements								Estimates for the entire ship		
Plates [kg]	Frames [kg]	Ice Stringers [kg]	Web Frames [kg]	Total [kg & t]		Weight difference to the lightweight option [t]	Weight increase from the lightweight option (only ice reinforcements considered)	Estimated total steel weight [t]	Weight of ice reinforcements divided by estimated total steel weight	Weight increase from the lightweight option
98468	37560	14974	22593	173594	173,59	3,20	1,88 %	1418,59	12,24 %	0,23 %
56,72 %	21,64 %	8,63 %	13,01 %							

Trans 350 & ice stringers										
Ice reinforcements								Estimates for the entire ship		
Plates [kg]	Frames [kg]	Ice Stringers [kg]	Web Frames [kg]	Total [kg & t]		Weight difference to the lightweight option [t]	Weight increase from the lightweight option (only ice reinforcements considered)	Estimated total steel weight [t]	Weight of ice reinforcements divided by estimated total steel weight	Weight increase from the lightweight option
78320	58993	17988	22581	177883	177,88	7,49	4,40 %	1422,88	12,50 %	0,53 %
44,03 %	33,16 %	10,11 %	12,69 %							

Table D1: Weight results for ice class IC in table format.

Long 550										
Ice reinforcements								Estimates for the entire ship		
Plates [kg]	Frames [kg]	Ice Stringers [kg]	Web Frames [kg]	Total [kg & t]		Weight difference to the lightweight option [t]	Weight increase from the lightweight option (only ice reinforcements considered)	Estimated total steel weight [t]	Weight of ice reinforcements divided by estimated total steel weight	Weight increase from the lightweight option
124854	52807	0	24493	202153	202,15	25,77	14,61 %	1447,15	13,97 %	1,81 %
61,76 %	26,12 %	0,00 %	12,12 %							

Long 650										
Ice reinforcements								Estimates for the entire ship		
Plates [kg]	Frames [kg]	Ice Stringers [kg]	Web Frames [kg]	Total [kg & t]		Weight difference to the lightweight option [t]	Weight increase from the lightweight option (only ice reinforcements considered)	Estimated total steel weight [t]	Weight of ice reinforcements divided by estimated total steel weight	Weight increase from the lightweight option
126840	44734	0	24518	196092	196,09	19,71	11,17 %	1441,09	13,61 %	1,39 %
64,68 %	22,81 %	0,00 %	12,50 %							

Long 750										
Ice reinforcements								Estimates for the entire ship		
Plates [kg]	Frames [kg]	Ice Stringers [kg]	Web Frames [kg]	Total [kg & t]		Weight difference to the lightweight option [t]	Weight increase from the lightweight option (only ice reinforcements considered)	Estimated total steel weight [t]	Weight of ice reinforcements divided by estimated total steel weight	Weight increase from the lightweight option
126194	35875	0	24518	186586	186,59	10,20	5,78 %	1431,59	13,03 %	0,72 %
67,63 %	19,23 %	0,00 %	13,14 %							

Long 650 - 375										
Ice reinforcements								Estimates for the entire ship		
Plates [kg]	Frames [kg]	Ice Stringers [kg]	Web Frames [kg]	Total [kg & t]		Weight difference to the lightweight option [t]	Weight increase from the lightweight option (only ice reinforcements considered)	Estimated total steel weight [t]	Weight of ice reinforcements divided by estimated total steel weight	Weight increase from the lightweight option
117349	58322	0	24438	200109	200,11	23,73	13,45 %	1445,11	13,85 %	1,67 %
58,64 %	29,15 %	0,00 %	12,21 %							

Trans 560										
Ice reinforcements								Estimates for the entire ship		
Plates [kg]	Frames [kg]	Ice Stringers [kg]	Web Frames [kg]	Total [kg & t]		Weight difference to the lightweight option [t]	Weight increase from the lightweight option (only ice reinforcements considered)	Estimated total steel weight [t]	Weight of ice reinforcements divided by estimated total steel weight	Weight increase from the lightweight option
96581	59441	0	24342	180364	180,36	3,98	2,26 %	1425,36	12,65 %	0,28 %
53,55 %	32,96 %	0,00 %	13,50 %							

Trans 560 & ice stringers										
Ice reinforcements								Estimates for the entire ship		
Plates [kg]	Frames [kg]	Ice Stringers [kg]	Web Frames [kg]	Total [kg & t]		Weight difference to the lightweight option [t]	Weight increase from the lightweight option (only ice reinforcements considered)	Estimated total steel weight [t]	Weight of ice reinforcements divided by estimated total steel weight	Weight increase from the lightweight option
96581	44816	14974	24342	180713	180,71	4,33	2,45 %	1425,71	12,68 %	0,30 %
53,44 %	24,80 %	8,29 %	13,47 %							

Trans 700										
Ice reinforcements								Estimates for the entire ship		
Plates [kg]	Frames [kg]	Ice Stringers [kg]	Web Frames [kg]	Total [kg & t]		Weight difference to the lightweight option [t]	Weight increase from the lightweight option (only ice reinforcements considered)	Estimated total steel weight [t]	Weight of ice reinforcements divided by estimated total steel weight	Weight increase from the lightweight option
99086	52956	0	24342	176384	176,38	0,00	0,00 %	1421,38	12,41 %	0,00 %
56,18 %	30,02 %	0,00 %	13,80 %							

Trans 700 & ice stringers										
Ice reinforcements								Estimates for the entire ship		
Plates [kg]	Frames [kg]	Ice Stringers [kg]	Web Frames [kg]	Total [kg & t]		Weight difference to the lightweight option [t]	Weight increase from the lightweight option (only ice reinforcements considered)	Estimated total steel weight [t]	Weight of ice reinforcements divided by estimated total steel weight	Weight increase from the lightweight option
99086	39336	14974	24342	177737	177,74	1,35	0,77 %	1422,74	12,49 %	0,10 %
55,75 %	22,13 %	8,42 %	13,70 %							

Trans 350 & ice stringers										
Ice reinforcements								Estimates for the entire ship		
Plates [kg]	Frames [kg]	Ice Stringers [kg]	Web Frames [kg]	Total [kg & t]		Weight difference to the lightweight option [t]	Weight increase from the lightweight option (only ice reinforcements considered)	Estimated total steel weight [t]	Weight of ice reinforcements divided by estimated total steel weight	Weight increase from the lightweight option
78644	61543	17988	24330	182505	182,51	6,12	3,47 %	1427,51	12,78 %	0,43 %
43,09 %	33,72 %	9,86 %	13,33 %							

Table D2: Weight results for ice class IB in table format.

Long 550										
Ice reinforcements								Estimates for the entire ship		
Plates [kg]	Frames [kg]	Ice Stringers [kg]	Web Frames [kg]	Total [kg & t]		Weight difference to the lightweight option [t]	Weight increase from the lightweight option (only ice reinforcements considered)	Estimated total steel weight [t]	Weight of ice reinforcements divided by estimated total steel weight	Weight increase from the lightweight option
163167	70435	0	28935	262536	262,54	52,65	25,08 %	1489,54	17,63 %	3,66 %
62,15 %	26,83 %	0,00 %	11,02 %							

Long 650										
Ice reinforcements								Estimates for the entire ship		
Plates [kg]	Frames [kg]	Ice Stringers [kg]	Web Frames [kg]	Total [kg & t]		Weight difference to the lightweight option [t]	Weight increase from the lightweight option (only ice reinforcements considered)	Estimated total steel weight [t]	Weight of ice reinforcements divided by estimated total steel weight	Weight increase from the lightweight option
165391	56787	0	28984	251162	251,16	41,27	19,66 %	1478,16	16,99 %	2,87 %
65,85 %	22,61 %	0,00 %	11,54 %							

Long 750										
Ice reinforcements								Estimates for the entire ship		
Plates [kg]	Frames [kg]	Ice Stringers [kg]	Web Frames [kg]	Total [kg & t]		Weight difference to the lightweight option [t]	Weight increase from the lightweight option (only ice reinforcements considered)	Estimated total steel weight [t]	Weight of ice reinforcements divided by estimated total steel weight	Weight increase from the lightweight option
167961	43883	0	28984	240828	240,83	30,94	14,74 %	1467,83	16,41 %	2,15 %
69,74 %	18,22 %	0,00 %	12,04 %							

Long 650 - 375										
Ice reinforcements								Estimates for the entire ship		
Plates [kg]	Frames [kg]	Ice Stringers [kg]	Web Frames [kg]	Total [kg & t]		Weight difference to the lightweight option [t]	Weight increase from the lightweight option (only ice reinforcements considered)	Estimated total steel weight [t]	Weight of ice reinforcements divided by estimated total steel weight	Weight increase from the lightweight option
151231	71119	0	28855	251205	251,21	41,32	19,68 %	1478,21	16,99 %	2,88 %
60,20 %	28,31 %	0,00 %	11,49 %							

Trans 560										
Ice reinforcements								Estimates for the entire ship		
Plates [kg]	Frames [kg]	Ice Stringers [kg]	Web Frames [kg]	Total [kg & t]		Weight difference to the lightweight option [t]	Weight increase from the lightweight option (only ice reinforcements considered)	Estimated total steel weight [t]	Weight of ice reinforcements divided by estimated total steel weight	Weight increase from the lightweight option
130199	66415	0	28782	225396	225,40	15,51	7,39 %	1452,40	15,52 %	1,08 %
57,76 %	29,47 %	0,00 %	12,77 %							

Trans 560 & ice stringers										
Ice reinforcements								Estimates for the entire ship		
Plates [kg]	Frames [kg]	Ice Stringers [kg]	Web Frames [kg]	Total [kg & t]		Weight difference to the lightweight option [t]	Weight increase from the lightweight option (only ice reinforcements considered)	Estimated total steel weight [t]	Weight of ice reinforcements divided by estimated total steel weight	Weight increase from the lightweight option
129121	48526	15529	28782	221958	221,96	12,07	5,75 %	1448,96	15,32 %	0,84 %
58,17 %	21,86 %	7,00 %	12,97 %							

Trans 700										
Ice reinforcements								Estimates for the entire ship		
Plates [kg]	Frames [kg]	Ice Stringers [kg]	Web Frames [kg]	Total [kg & t]		Weight difference to the lightweight option [t]	Weight increase from the lightweight option (only ice reinforcements considered)	Estimated total steel weight [t]	Weight of ice reinforcements divided by estimated total steel weight	Weight increase from the lightweight option
133798	60236	0	28782	222817	222,82	12,93	6,16 %	1449,82	15,37 %	0,90 %
60,05 %	27,03 %	0,00 %	12,92 %							

Trans 700 & ice stringers										
Ice reinforcements								Estimates for the entire ship		
Plates [kg]	Frames [kg]	Ice Stringers [kg]	Web Frames [kg]	Total [kg & t]		Weight difference to the lightweight option [t]	Weight increase from the lightweight option (only ice reinforcements considered)	Estimated total steel weight [t]	Weight of ice reinforcements divided by estimated total steel weight	Weight increase from the lightweight option
132720	43266	15529	28782	220298	220,30	10,41	4,96 %	1447,30	15,22 %	0,72 %
60,25 %	19,64 %	7,05 %	13,07 %							

Trans 350 & ice stringers										
Ice reinforcements								Estimates for the entire ship		
Plates [kg]	Frames [kg]	Ice Stringers [kg]	Web Frames [kg]	Total [kg & t]		Weight difference to the lightweight option [t]	Weight increase from the lightweight option (only ice reinforcements considered)	Estimated total steel weight [t]	Weight of ice reinforcements divided by estimated total steel weight	Weight increase from the lightweight option
97025	66173	17933	28759	209890	209,89	0,00	0,00 %	1436,89	14,61 %	0,00 %
46,23 %	31,53 %	8,54 %	13,70 %							

Table D3: Weight results for ice class IA in table format.

Long 550										
Ice reinforcements								Estimates for the entire ship		
Plates [kg]	Frames [kg]	Ice Stringers [kg]	Web Frames [kg]	Total [kg & t]		Weight difference to the lightweight option [t]	Weight increase from the lightweight option (only ice reinforcements considered)	Estimated total steel weight [t]	Weight of ice reinforcements divided by estimated total steel weight	Weight increase from the lightweight option
243879	113185	0	33479	390543	390,54	138,21	54,78 %	1583,54	24,66 %	9,56 %
62,45 %	28,98 %	0,00 %	8,57 %				9,57 %			

Long 650										
Ice reinforcements								Estimates for the entire ship		
Plates [kg]	Frames [kg]	Ice Stringers [kg]	Web Frames [kg]	Total [kg & t]		Weight difference to the lightweight option [t]	Weight increase from the lightweight option (only ice reinforcements considered)	Estimated total steel weight [t]	Weight of ice reinforcements divided by estimated total steel weight	Weight increase from the lightweight option
245380	79701	0	33513	358594	358,59	106,26	42,11 %	1551,59	23,11 %	7,35 %
68,43 %	22,23 %	0,00 %	9,35 %				7,35 %			

Long 750										
Ice reinforcements								Estimates for the entire ship		
Plates [kg]	Frames [kg]	Ice Stringers [kg]	Web Frames [kg]	Total [kg & t]		Weight difference to the lightweight option [t]	Weight increase from the lightweight option (only ice reinforcements considered)	Estimated total steel weight [t]	Weight of ice reinforcements divided by estimated total steel weight	Weight increase from the lightweight option
252368	69468	0	33513	355349	355,35	103,02	40,83 %	1548,35	22,95 %	7,13 %
71,02 %	19,55 %	0,00 %	9,43 %				7,13 %			

Long 650 - 375										
Ice reinforcements								Estimates for the entire ship		
Plates [kg]	Frames [kg]	Ice Stringers [kg]	Web Frames [kg]	Total [kg & t]		Weight difference to the lightweight option [t]	Weight increase from the lightweight option (only ice reinforcements considered)	Estimated total steel weight [t]	Weight of ice reinforcements divided by estimated total steel weight	Weight increase from the lightweight option
224995	96872	0	33404	355270	355,27	102,94	40,80 %	1548,27	22,95 %	7,12 %
63,33 %	27,27 %	0,00 %	9,40 %				7,12 %			

Trans 560										
Ice reinforcements								Estimates for the entire ship		
Plates [kg]	Frames [kg]	Ice Stringers [kg]	Web Frames [kg]	Total [kg & t]		Weight difference to the lightweight option [t]	Weight increase from the lightweight option (only ice reinforcements considered)	Estimated total steel weight [t]	Weight of ice reinforcements divided by estimated total steel weight	Weight increase from the lightweight option
193516	78010	0	33315	304841	304,84	52,51	20,81 %	1497,84	20,35 %	3,63 %
63,48 %	25,59 %	0,00 %	10,93 %				3,63 %			

Trans 560 & ice stringers										
Ice reinforcements								Estimates for the entire ship		
Plates [kg]	Frames [kg]	Ice Stringers [kg]	Web Frames [kg]	Total [kg & t]		Weight difference to the lightweight option [t]	Weight increase from the lightweight option (only ice reinforcements considered)	Estimated total steel weight [t]	Weight of ice reinforcements divided by estimated total steel weight	Weight increase from the lightweight option
193516	55900	17124	33315	299855	299,85	47,53	18,84 %	1492,85	20,09 %	3,29 %
64,54 %	18,64 %	5,71 %	11,11 %				3,29 %			

Trans 700										
Ice reinforcements								Estimates for the entire ship		
Plates [kg]	Frames [kg]	Ice Stringers [kg]	Web Frames [kg]	Total [kg & t]		Weight difference to the lightweight option [t]	Weight increase from the lightweight option (only ice reinforcements considered)	Estimated total steel weight [t]	Weight of ice reinforcements divided by estimated total steel weight	Weight increase from the lightweight option
203938	72010	0	33315	309263	309,26	56,93	22,56 %	1502,26	20,59 %	3,94 %
65,94 %	23,28 %	0,00 %	10,77 %				3,94 %			

Trans 700 & ice stringers										
Ice reinforcements								Estimates for the entire ship		
Plates [kg]	Frames [kg]	Ice Stringers [kg]	Web Frames [kg]	Total [kg & t]		Weight difference to the lightweight option [t]	Weight increase from the lightweight option (only ice reinforcements considered)	Estimated total steel weight [t]	Weight of ice reinforcements divided by estimated total steel weight	Weight increase from the lightweight option
203938	52917	17507	33315	307677	307,68	55,35	21,94 %	1500,68	20,50 %	3,83 %
66,28 %	17,20 %	5,69 %	10,83 %				3,83 %			

Trans 350 & ice stringers										
Ice reinforcements								Estimates for the entire ship		
Plates [kg]	Frames [kg]	Ice Stringers [kg]	Web Frames [kg]	Total [kg & t]		Weight difference to the lightweight option [t]	Weight increase from the lightweight option (only ice reinforcements considered)	Estimated total steel weight [t]	Weight of ice reinforcements divided by estimated total steel weight	Weight increase from the lightweight option
132570	69589	16893	33276	252329	252,33	0,00	0,00 %	1445,33	17,46 %	0,00 %
52,54 %	27,58 %	6,69 %	13,19 %				0,00 %			

Table D4: Weight results for ice class IA Super in table format.

E Source code for the Python program

```python
#Created on 15.6.2020

#Author: Tuomas Avellan
#Program developed for Master's thesis study.

import os
import math
import xlrd
import xlsxwriter
import copy

def main():
    ice_class, framing_sys, min_power, displacement, shell_plate_area_lst, frame_lst, stringer_lst, web_lst, bulb_lst, T_bar_lst, shell_plates_lst = input()
    effective_breadth =  0.650
    sigma_y = 315
    web_frame_spacing = 2.8
    frame_spacing = 1.4
    selected_shell_plates_lst = shell_plating_function(ice_class,min_power,displacement,framing_sys,frame_lst,shell_plates_lst,sigma_y)
    if(framing_sys == "transverse"):
        frames_final = transverse_frame_function(ice_class, min_power, displacement, frame_lst, bulb_lst, T_bar_lst, selected_shell_plates_lst, effective_breadth, sigma_y)
    elif(framing_sys == "longitudinal"):
        frames_final = longitudinal_frame_function(ice_class, min_power, displacement, frame_lst, bulb_lst, T_bar_lst, selected_shell_plates_lst, effective_breadth, sigma_y)
    if(stringer_lst == False):
        stringers_final = False
    else:
        stringers_final = ice_stringer_function(ice_class, min_power, displacement, stringer_lst, framing_sys, T_bar_lst, selected_shell_plates_lst, effective_breadth, sigma_y)
    webs_final = web_frame_function(ice_class, min_power, displacement, web_lst, framing_sys, T_bar_lst, selected_shell_plates_lst, effective_breadth, sigma_y)
    output(frames_final,stringers_final,webs_final,shell_plate_area_lst,selected_shell_plates_lst)



def input():
    file_path = "D:\\CASES\\Project\\DEferry_TA_Sandbox\\manual_results\\baseline\\IceReinforcementOpt\\IR_input.txt"
    with open(file_path,'r') as f:
        f.seek(0)
        ice_class = float(f.readline())
        fr_sys = int(f.readline())
        if (fr_sys == 1):
            framing_system = "transverse"
        else:
            framing_system = "longitudinal"

        min_power = float(f.readline())
        displacement = float(f.readline())

        shell_plate_area_lst = []
        line = f.readline()
        area_temp = line.replace("[","")
        area_temp2 = area_temp.replace("]","")
        area_lst_temp = area_temp2.split(",")
        for i in area_lst_temp:
            shell_plate_area_lst.append(float(i))
```

```

    stringer_lst = []
    frame_lst = []
    web_lst = []

    line = f.readline()
    temp_lst = line.split("]], [")
    for i in temp_lst:
        temp_lst2 = i.split("], [")
        for e in temp_lst2:
            text_temp = e.replace("[","")

            text_temp2 = text_temp.replace("]","")
            temp_lst2 = text_temp2.split(",")
            if len(temp_lst2) == 9:
                inner_lst =[]
                inner_lst.append(int(temp_lst2[0]))
                inner_lst.append(float(temp_lst2[1]))
                inner_lst.append([float(temp_lst2[2]),float(temp_lst2[3]),float(temp_lst2[4]),float(temp_lst2[5]),float(temp_lst2[6]),float(temp_lst2[7]),float(temp_lst2[8])])
                temp_lst2.pop(0)
            else:
                inner_lst2 = [float(u) for u in temp_lst2]
                inner_lst.append(inner_lst2)

        frame_lst.append(inner_lst)

    line = f.readline()
    temp_lst = line.split("]], [")
    if(len(temp_lst) == 1):
        stringer_lst = False
    else:
        for i in temp_lst:
            temp_lst2  = i.split("], [")
            for e in temp_lst2:
                text_temp = e.replace("[","")

                text_temp2 = text_temp.replace("]","")
                temp_lst2 = text_temp2.split(",")
                if len(temp_lst2) == 8:
                    inner_lst =[]
                    inner_lst.append(int(temp_lst2[0]))
                    inner_lst.append(float(temp_lst2[1]))
                    inner_lst.append([float(temp_lst2[2]),float(temp_lst2[3]),float(temp_lst2[4]),float(temp_lst2[5]),float(temp_lst2[6]),float(temp_lst2[7])])
                    temp_lst2.pop(0)
                else:
                    inner_lst2 = [float(u) for u in temp_lst2]
                    inner_lst.append(inner_lst2)

            stringer_lst.append(inner_lst)

    line = f.readline()
    temp_lst = line.split("]], [")
    for i in temp_lst:
        temp_lst2 = i.split("], [")
        for e in temp_lst2:
            text_temp = e.replace("[","")

            text_temp2 = text_temp.replace("]","")
            temp_lst2 = text_temp2.split(",")
            if len(temp_lst2) == 9:
                inner_lst =[]
                inner_lst.append(int(temp_lst2[0]))
                inner_lst.append(float(temp_lst2[1]))
```

```python
                    inner_lst.append([float(temp_lst2[2]),float(temp_lst2[3]),float(temp_lst2[4]),float(temp_lst2[5]),float(temp_lst2[6]),float(temp_lst2[7]),float(temp_lst2[8])])
                    temp_lst2.pop(0)
                else:
                    inner_lst2 = [float(u) for u in temp_lst2]
                    inner_lst.append(inner_lst2)

            web_lst.append(inner_lst)

    file_path = "D:\\CASES\\Project\\DEferry_TA_Sandbox\\manual_results\\baseline\\IceReinforcementOpt\\Steel_profile_lib.xlsx"
    bulb_profiles = []
    T_bar_profiles = []
    Shell_plates = []
    i = 3
    libraryxlsx = xlrd.open_workbook(file_path)
    worksheet = libraryxlsx.sheet_by_index(0)


    while (i < worksheet.nrows):
        bulb_profiles.append([str(worksheet.cell_value(i,1)),float(worksheet.cell_value(i,2)),float(worksheet.cell_value(i,3)),float(worksheet.cell_value(i,4)),float(worksheet.cell_value(i,5)),float(worksheet.cell_value(i,6)),float(worksheet.cell_value(i,7)),float(worksheet.cell_value(i,8)),float(worksheet.cell_value(i,9)),float(worksheet.cell_value(i,10)),float(worksheet.cell_value(i,11)),float(worksheet.cell_value(i,12))])
        i += 1

    worksheet = libraryxlsx.sheet_by_index(1)
    i = 3
    while (i < worksheet.nrows):
        T_bar_profiles.append([str(worksheet.cell_value(i,1)),float(worksheet.cell_value(i,2)),float(worksheet.cell_value(i,3)),float(worksheet.cell_value(i,4)),float(worksheet.cell_value(i,5)),float(worksheet.cell_value(i,6)),float(worksheet.cell_value(i,7)),float(worksheet.cell_value(i,8)),float(worksheet.cell_value(i,9)),float(worksheet.cell_value(i,10)),float(worksheet.cell_value(i,11)),float(worksheet.cell_value(i,12))])
        i += 1

    worksheet = libraryxlsx.sheet_by_index(2)
    i = 3
    while (i < worksheet.nrows):
        Shell_plates.append(float(worksheet.cell_value(i,1)))
        i += 1

    return(ice_class, framing_system, min_power, displacement, shell_plate_area_lst, frame_lst, stringer_lst, web_lst, bulb_profiles, T_bar_profiles, Shell_plates)

def output(frames_final,stringers_final,webs_final,shell_plate_area_lst, selected_shell_plates_lst):
    file_path = "D:\\CASES\\Project\\DEferry_TA_Sandbox\\manual_results\\baseline\\IceReinforcementOpt\\Output.xlsx"
    outWorkbook = xlsxwriter.Workbook(file_path)
    special_cell_format = outWorkbook.add_format({"font_color":"red"})
    outSheet = outWorkbook.add_worksheet("Plates")
    outSheet.write("B2","Area index")
    outSheet.write("C2","Area (m^2)")
    outSheet.write("D2","Shell thickness (mm)")
    outSheet.write("E2","Weight (kg)")
    i = 2
    e = 1
    total_IB_shell_weight = 0
    for item in shell_plate_area_lst:
        outSheet.write(i,1,e)
        outSheet.write(i,2,item)
        outSheet.write(i,3,selected_shell_plates_lst[e])
```

```
        shell_weight = selected_shell_plates_lst[e]*(item/1000)*7800
        total_IB_shell_weight += shell_weight
        outSheet.write(i,4,shell_weight)
        i +=1
        e += 1

    outSheet.write("J2","IB Shell Plating Weight (1/4) [kg]")
    outSheet.write("J3",total_IB_shell_weight)
    outSheet.write("J6","Total IB Shell Plating Weight [kg]")
    outSheet.write("J7",total_IB_shell_weight*4)


    outSheet = outWorkbook.add_worksheet("Ice Frames")
    outSheet.write("B2","Ice Frame")
    outSheet.write("C2","Location")
    outSheet.write("D2","Frame Weight [kg]")
    outSheet.write("E2","Part Name")
    outSheet.write("F2","Part Span [m]")
    outSheet.write("G2","Part weights [kg]")
    i = 2
    e = i
    total_frame_weight = 0
    frame_weight = 0
    parts_outside_IR_weight = 0
    for item in frames_final:
        u = 0
        if i > 2:
            outSheet.write(e,3, frame_weight)
            total_frame_weight += frame_weight
            frame_weight = 0
            i += 2
            e = i
        outSheet.write(i,1, item[0][0])
        outSheet.write(i,2, item[0][1])
        for frame_item in item:
            if u > 0:
                outSheet.write(i,4, frame_item[0])
                outSheet.write(i,5, frame_item[2])
                outSheet.write(i,6, frame_item[3])
                if len(frame_item) == 5:
                    parts_outside_IR_weight +=frame_item[3]
                    outSheet.write(i,7, False,special_cell_format)

                frame_weight += frame_item[3]
            u += 1
            i += 1

    outSheet.write(e,3, frame_weight)
    total_frame_weight += frame_weight

    outSheet.write("J2","Ice Frame Weight (1/4)")
    outSheet.write("J3",total_frame_weight)
    outSheet.write("J6","Total Ice Frame Weight [kg]")
    outSheet.write("J7",total_frame_weight*4)
    outSheet.write("J10","(Weight from frame parts outside IR) / (Total Ice Frame Weight)")
    outSheet.write("J11",parts_outside_IR_weight/total_frame_weight)

    if(stringers_final==False):
        outSheet = outWorkbook.add_worksheet("Stringers")
        outSheet.write("B2","No Ice Stringers")
        total_stringer_weight = 0
    else:
        outSheet = outWorkbook.add_worksheet("Stringers")
        outSheet.write("B2","Stringer")
        outSheet.write("C2","Location")
        outSheet.write("D2","Weight [kg]")
        outSheet.write("E2","Part")
```

```
        outSheet.write("F2","Part Span [m]")
        outSheet.write("G2","Part weight [kg]")
        i = 2
        e = i
        total_stringer_weight = 0
        stringer_weight = 0
        for item in stringers_final:
            u = 0
            if i > 2:
                outSheet.write(e,3, stringer_weight)
                total_stringer_weight += stringer_weight
                stringer_weight = 0
                i += 2
                e = i
            outSheet.write(i,1, item[0][0])
            outSheet.write(i,2, item[0][1])
            for stringer_item in item:
                if u > 0:
                    outSheet.write(i,4, stringer_item[0])
                    outSheet.write(i,5, stringer_item[2])
                    outSheet.write(i,6, stringer_item[3])
                    stringer_weight += stringer_item[3]
                u += 1
                i += 1

        outSheet.write(e,3, stringer_weight)
        total_stringer_weight += stringer_weight

        outSheet.write("J2","Ice Stringer Weight (1/4)")
        outSheet.write("J3",total_stringer_weight)
        outSheet.write("J6","Total Ice Stringer Weight [kg]")
        outSheet.write("J7",total_stringer_weight*4)

    outSheet = outWorkbook.add_worksheet("Web Frames")
    outSheet.write("B2","Web Frame")
    outSheet.write("C2","Location")
    outSheet.write("D2","Weight [kg]")
    outSheet.write("E2","Part")
    outSheet.write("F2","Part Span [m]")
    outSheet.write("G2","Part weight [kg]")
    i = 2
    e = i
    total_web_weight = 0
    web_weight = 0
    for item in webs_final:
        u = 0
        if i > 2:
            outSheet.write(e,3, web_weight)
            total_web_weight += web_weight
            web_weight = 0
            i += 2
            e = i
        outSheet.write(i,1, item[0][0])
        outSheet.write(i,2, item[0][1])
        for web_item in item:
            if u > 0:
                outSheet.write(i,4, web_item[0])
                outSheet.write(i,5, web_item[2])
                outSheet.write(i,6, web_item[3])
                web_weight += web_item[3]
            u += 1
            i += 1

    outSheet.write(e,3, web_weight)
    total_web_weight += web_weight

    outSheet.write("J2","Web Frame Weight (1/4)")
    outSheet.write("J3",total_web_weight)
```

```
    outSheet.write("J6","Total Web Frame Weight [kg]")
    outSheet.write("J7",total_web_weight*4)

    outSheet = outWorkbook.add_worksheet("Overview")
    outSheet.write("B2","Plates [kg]")
    outSheet.write("C2","Ice Frames [kg]")
    outSheet.write("D2","Ice Stringers [kg]")
    outSheet.write("E2","Web Frames [kg]")
    outSheet.write("G2","Total [kg]")
    outSheet.write("K6","Ice Frames outside Ice Region / Ice Frames")
    outSheet.write("K9","Ice Frames outside Ice Region / Total")
    outSheet.write("B3",total_IB_shell_weight*4)
    outSheet.write("B4",(total_IB_shell_weight*4)/((total_IB_shell_weight+total_frame_weight+total_stringer_weight+total_web_weight)*4))
    outSheet.write("C3",total_frame_weight*4)
    outSheet.write("C4",(total_frame_weight*4)/((total_IB_shell_weight+total_frame_weight+total_stringer_weight+total_web_weight)*4))
    outSheet.write("D3",total_stringer_weight*4)
    outSheet.write("D4",(total_stringer_weight*4)/((total_IB_shell_weight+total_frame_weight+total_stringer_weight+total_web_weight)*4))
    outSheet.write("E3",total_web_weight*4)
    outSheet.write("E4",(total_web_weight*4)/((total_IB_shell_weight+total_frame_weight+total_stringer_weight+total_web_weight)*4))
    outSheet.write("G3",(total_IB_shell_weight+total_frame_weight+total_stringer_weight+total_web_weight)*4)
    outSheet.write("K7",parts_outside_IR_weight/total_frame_weight)
    outSheet.write("K10",(parts_outside_IR_weight*4)/((total_IB_shell_weight+total_frame_weight+total_stringer_weight+total_web_weight)*4))
    outWorkbook.close()

    file_path = "D:\\CASES\\Project\\DEferry_TA_Sandbox\\manual_results\\baseline\\IceReinforcementOpt\\IR_output.txt"
    with open(file_path,’w’) as f:
        Total_weight = (total_IB_shell_weight+total_frame_weight+total_stringer_weight+total_web_weight)*4
        printti = str(Total_weight)
        f.write("weight = " + printti+ "\n")
    print("Done")

def shell_plating_function(ice_class,min_power,displacement,framing_sys,frame_lst,shell_plates_lst,sigma_y):
    frame_lst_copy = copy.deepcopy(frame_lst)

    plate_1_frame = [0,0,0,0,0,0,0]
    plate_2_frame = [0,0,0,0,0,0,0]
    plate_3_frame = [0,0,0,0,0,0,0]
    plate_4_frame = [0,0,0,0,0,0,0]
    plate_5_frame = [0,0,0,0,0,0,0]
    plate_6_frame = [0,0,0,0,0,0,0]
    plate_7_frame = [0,0,0,0,0,0,0]
    plate_8_frame = [0,0,0,0,0,0,0]
    plate_9_frame = [0,0,0,0,0,0,0]
    plate_10_frame = [0,0,0,0,0,0,0]
    plate_11_frame = [0,0,0,0,0,0,0]
    plate_12_frame = [0,0,0,0,0,0,0]

    for fr in frame_lst_copy:
        fr.pop(0)
        fr.pop(0)
        for fr_part in fr:

            if(fr_part[3] == 1):
                if(plate_1_frame[6] < fr_part[6]):
                    plate_1_frame = fr_part
            elif(fr_part[3] == 2):
                if(plate_2_frame[6] < fr_part[6]):
                    plate_2_frame = fr_part
            elif(fr_part[3] == 3):
```

```
            if(plate_3_frame[6] < fr_part[6]):
                plate_3_frame = fr_part
        elif(fr_part[3] == 4):
            if(plate_4_frame[6] < fr_part[6]):
                plate_4_frame = fr_part
        elif(fr_part[3] == 5):
            if(plate_5_frame[6] < fr_part[6]):
                plate_5_frame = fr_part
        elif(fr_part[3] == 6):
            if(plate_6_frame[6] < fr_part[6]):
                plate_6_frame = fr_part
        elif(fr_part[3] == 7):
            if(plate_7_frame[6] < fr_part[6]):
                plate_7_frame = fr_part
        elif(fr_part[3] == 8):
            if(plate_8_frame[6] < fr_part[6]):
                plate_8_frame = fr_part
        elif(fr_part[3] == 9):
            if(plate_9_frame[6] < fr_part[6]):
                plate_9_frame = fr_part
        elif(fr_part[3] == 10):
            if(plate_10_frame[6] < fr_part[6]):
                plate_10_frame = fr_part
        elif(fr_part[3] == 11):
            if(plate_11_frame[6] < fr_part[6]):
                plate_11_frame = fr_part
        elif(fr_part[3] == 12):
            if(plate_12_frame[6] < fr_part[6]):
                plate_12_frame = fr_part


shell_2 = shell_plate_thickness(ice_class,displacement,min_power,"mid",framing_sys,plate_2_frame[5],plate_2_frame[6],shell_plates_lst,sigma_y)
if(plate_1_frame[6] == 0):
    shell_1 = shell_2
else:
    shell_1 = shell_plate_thickness(ice_class,displacement,min_power,"mid",framing_sys,plate_1_frame[5],plate_1_frame[6],shell_plates_lst,sigma_y)

shell_3 = shell_plate_thickness(ice_class,displacement,min_power,"bow",framing_sys,plate_3_frame[5],plate_3_frame[6],shell_plates_lst,sigma_y)
shell_4 = shell_plate_thickness(ice_class,displacement,min_power,"bow",framing_sys,plate_4_frame[5],plate_4_frame[6],shell_plates_lst,sigma_y)
if(plate_5_frame[6] == 0 or plate_5_frame[6] > 0.4):
    shell_5 = shell_plate_thickness(ice_class,displacement,min_power,"bow","transverse",0,0.715,shell_plates_lst,sigma_y)
else:
    shell_5 = shell_plate_thickness(ice_class,displacement,min_power,"bow",framing_sys,plate_5_frame[5],plate_5_frame[6],shell_plates_lst,sigma_y)

shell_6 = shell_plate_thickness(ice_class,displacement,min_power,"bow",framing_sys,plate_6_frame[5],plate_6_frame[6],shell_plates_lst,sigma_y)
if(plate_7_frame[6] == 0 or plate_7_frame[6] > 0.4):
    shell_7 = shell_plate_thickness(ice_class,displacement,min_power,"bow","transverse",0,0.715,shell_plates_lst,sigma_y)
else:
    shell_7 = shell_plate_thickness(ice_class,displacement,min_power,"bow",framing_sys,plate_7_frame[5],plate_7_frame[6],shell_plates_lst,sigma_y)

shell_8 = shell_plate_thickness(ice_class,displacement,min_power,"bow",framing_sys,plate_8_frame[5],plate_8_frame[6],shell_plates_lst,sigma_y)
if(plate_9_frame[6] == 0 or plate_9_frame[6] > 0.4):
    shell_9 = shell_plate_thickness(ice_class,displacement,min_power,"bow","transverse",0,0.715,shell_plates_lst,sigma_y)
else:
    shell_9 = shell_plate_thickness(ice_class,displacement,min_power,"bow",framing_sys,plate_9_frame[5],plate_9_frame[6],shell_plates_lst,sigma_y)

```

```
    shell_10 = shell_plate_thickness(ice_class,displacement,min_power,"bow",framing_sys,plate_10_frame[5],plate_10_frame[6],shell_plates_lst,sigma_y)
    if(plate_11_frame[6] == 0 or plate_11_frame[6] > 0.4):
        shell_11 = shell_plate_thickness(ice_class,displacement,min_power,"bow","transverse",0,0.715,shell_plates_lst,sigma_y)
    else:
        shell_11 = shell_plate_thickness(ice_class,displacement,min_power,"bow",framing_sys,plate_11_frame[5],plate_11_frame[6],shell_plates_lst,sigma_y)

    shell_12 = shell_plate_thickness(ice_class,displacement,min_power,"bow",framing_sys,plate_12_frame[5],plate_12_frame[6],shell_plates_lst,sigma_y)

    chosen_shell_thickness_lst =[16,shell_1,shell_2,shell_3,shell_4,shell_5,shell_6,shell_7,shell_8,shell_9,shell_10,shell_11,shell_12]
    return(chosen_shell_thickness_lst)


def transverse_frame_function(ice_class, min_power, displacement, frame_lst, bulb_lst,T_bar_lst, selected_shell_plates_lst, effective_breadth, sigma_y):
    structure_type = "frame"
    framing_sys = "transverse"
    number_of_frames = len(frame_lst)
    frames_final =[]

    for fr in frame_lst:
        frame_index = fr[0]
        frame_x_cord = fr[1]
        calculated_frame = [[frame_index,frame_x_cord]]
        fr.pop(0)
        fr.pop(0)

        for fr_part in fr:

            if(fr_part[0] == 1.0):
                location = "bow"
            elif(fr_part[0] == 2.0):
                location = "mid"
            else:
                location = "aft"

            if(fr_part[1] == 1.0):
                ice_reinforced = True
            elif(fr_part[1] == 0.0):
                ice_reinforced = False

            if(fr_part[2] == 1 or fr_part[3] == 0):
                plate_thickness = selected_shell_plates_lst[0]
            elif(fr_part[3] == 1):
                plate_thickness = selected_shell_plates_lst[1]
            elif(fr_part[3] == 2):
                plate_thickness = selected_shell_plates_lst[2]
            elif(fr_part[3] == 3):
                plate_thickness = selected_shell_plates_lst[3]
            elif(fr_part[3] == 4):
                plate_thickness = selected_shell_plates_lst[4]
            elif(fr_part[3] == 5):
                plate_thickness = selected_shell_plates_lst[5]
            elif(fr_part[3] == 6):
                plate_thickness = selected_shell_plates_lst[6]
            elif(fr_part[3] == 7):
                plate_thickness = selected_shell_plates_lst[7]
            elif(fr_part[3] == 8):
                plate_thickness = selected_shell_plates_lst[8]
            elif(fr_part[3] == 9):
                plate_thickness = selected_shell_plates_lst[9]
            elif(fr_part[3] == 10):
                plate_thickness = selected_shell_plates_lst[10]
            elif(fr_part[3] == 11):
```

```
                plate_thickness = selected_shell_plates_lst[11]
            elif(fr_part[3] == 12):
                plate_thickness = selected_shell_plates_lst[12]

            true_frame_span = fr_part[4]
            frame_span = fr_part[5]
            frame_spacing = fr_part[6]

            if (frame_spacing < effective_breadth):
                effective_breadth = frame_spacing

            if (ice_reinforced == True):
                Z_requirement,A_requirement = frame_requirements(ice_class,displacement,min_power,location,framing_sys,frame_span,frame_spacing,sigma_y)

                profile, weight = bulb_profile_selection(plate_thickness,effective_breadth,Z_requirement,A_requirement, bulb_lst,T_bar_lst, sigma_y)

                mass = weight*true_frame_span

                calculated_part = [profile,weight,true_frame_span,mass]
            else:
                Z_requirement,A_requirement = frame_requirements(4,displacement,min_power,"aft",framing_sys,frame_span,frame_spacing,sigma_y)
                profile, weight = bulb_profile_selection(plate_thickness,effective_breadth,Z_requirement,A_requirement, bulb_lst,T_bar_lst, sigma_y)
                mass = weight*true_frame_span
                calculated_part = [profile,weight,true_frame_span,mass,False]
            calculated_frame.append(calculated_part)
        frames_final.append(calculated_frame)
    return(frames_final)

def longitudinal_frame_function(ice_class, min_power, displacement, frame_lst, bulb_lst, T_bar_lst, selected_shell_plates_lst, effective_breadth, sigma_y):
    structure_type = "frame"
    framing_sys = "longitudinal"
    number_of_frames = len(frame_lst)
    frames_final =[]

    for fr in frame_lst:

        frame_index = fr[0]
        frame_z_cord = fr[1]
        calculated_frame = [[frame_index,frame_z_cord]]
        fr.pop(0)
        fr.pop(0)

        for fr_part in fr:
            if(fr_part[0] == 1.0):
                location = "bow"
            elif(fr_part[0] == 2.0):
                location = "mid"
            else:
                location = "aft"

            if(fr_part[1] == 1.0):
                ice_reinforced = True
            elif(fr_part[1] == 0.0):
                ice_reinforced = False

            if(fr_part[2] == 1 or fr_part[3] == 0):
                plate_thickness = selected_shell_plates_lst[0]
            elif(fr_part[3] == 1):
                plate_thickness = selected_shell_plates_lst[1]
            elif(fr_part[3] == 2):
                plate_thickness = selected_shell_plates_lst[2]
            elif(fr_part[3] == 3):
                plate_thickness = selected_shell_plates_lst[3]
```

```
            elif(fr_part[3] == 4):
                plate_thickness = selected_shell_plates_lst[4]
            elif(fr_part[3] == 5):
                plate_thickness = selected_shell_plates_lst[5]
            elif(fr_part[3] == 6):
                plate_thickness = selected_shell_plates_lst[6]
            elif(fr_part[3] == 7):
                plate_thickness = selected_shell_plates_lst[7]
            elif(fr_part[3] == 8):
                plate_thickness = selected_shell_plates_lst[8]
            elif(fr_part[3] == 9):
                plate_thickness = selected_shell_plates_lst[9]
            elif(fr_part[3] == 10):
                plate_thickness = selected_shell_plates_lst[10]
            elif(fr_part[3] == 11):
                plate_thickness = selected_shell_plates_lst[11]
            elif(fr_part[3] == 12):
                plate_thickness = selected_shell_plates_lst[12]

            true_frame_span = fr_part[4]
            frame_span = fr_part[5]
            frame_spacing = fr_part[6]

            if (frame_spacing < effective_breadth):
                effective_breadth = frame_spacing

            if (ice_reinforced == True):
                Z_requirement,A_requirement = frame_requirements(ice_class,displacement,min_power,location,framing_sys,frame_span,frame_spacing,sigma_y)
                profile, weight = bulb_profile_selection(plate_thickness,effective_breadth,Z_requirement,A_requirement, bulb_lst, T_bar_lst, sigma_y)
                mass = weight*true_frame_span
                calculated_part = [profile,weight,true_frame_span,mass]
            else:
                Z_requirement,A_requirement = frame_requirements(4,displacement,min_power,"aft",framing_sys,frame_span,frame_spacing,sigma_y)
                profile, weight = bulb_profile_selection(plate_thickness,effective_breadth,Z_requirement,A_requirement, bulb_lst, T_bar_lst, sigma_y)
                mass = weight*true_frame_span
                calculated_part = [profile,weight,true_frame_span,mass,False]
            calculated_frame.append(calculated_part)
        frames_final.append(calculated_frame)
    return(frames_final)

def ice_stringer_function(ice_class,min_power,displacement,stringer_lst,framing_sys, T_bar_lst, selected_shell_plates_lst, effective_breadth, sigma_y):

    structure_type = "stringer"
    number_of_ice_stringers = len(stringer_lst)
    stringers_final =[]
    for stringer in stringer_lst:
        stringer_index = stringer[0]
        stringer_z_cord = stringer[1]
        calculated_stringer = [[stringer_index,stringer_z_cord]]
        stringer.pop(0)
        stringer.pop(0)
        for stringer_part in stringer:
            if (stringer_part[0]==1.0):
                location = "bow"
            elif (stringer_part[0] == 2.0):
                location = "mid"
            else:
                location = "aft"

            within_ib = True
            if(stringer_part[1] == 0):
                plate_thickness = selected_shell_plates_lst[0]
                within_ib = False
```

```
            elif(stringer_part[1] == 1):
                plate_thickness = selected_shell_plates_lst[1]
            elif(stringer_part[1] == 2):
                plate_thickness = selected_shell_plates_lst[2]
            elif(stringer_part[1] == 3):
                plate_thickness = selected_shell_plates_lst[3]
            elif(stringer_part[1] == 4):
                plate_thickness = selected_shell_plates_lst[4]
            elif(stringer_part[1] == 5):
                plate_thickness = selected_shell_plates_lst[5]
            elif(stringer_part[1] == 6):
                plate_thickness = selected_shell_plates_lst[6]
            elif(stringer_part[1] == 7):
                plate_thickness = selected_shell_plates_lst[7]
            elif(stringer_part[1] == 8):
                plate_thickness = selected_shell_plates_lst[8]
            elif(stringer_part[1] == 9):
                plate_thickness = selected_shell_plates_lst[9]
            elif(stringer_part[1] == 10):
                plate_thickness = selected_shell_plates_lst[10]
            elif(stringer_part[1] == 11):
                plate_thickness = selected_shell_plates_lst[11]
            elif(stringer_part[1] == 12):
                plate_thickness = selected_shell_plates_lst[12]

            true_stringer_span = stringer_part[2]
            stringer_span = stringer_part[3]
            stringer_spacing = stringer_part[4]
            stringer_distance_to_ib = stringer_part[5]
            Z_requirement,A_requirement = stringer_requirements(ice_class,
    displacement, min_power, location, within_ib, framing_sys, stringer_span,
    stringer_spacing, stringer_distance_to_ib, sigma_y)
            profile, weight = T_bar_profile_selection(plate_thickness,
    effective_breadth,Z_requirement,A_requirement, T_bar_lst, sigma_y)
            mass = weight*true_stringer_span
            calculated_part = [profile,weight,true_stringer_span,mass]
            calculated_stringer.append(calculated_part)
        stringers_final.append(calculated_stringer)
    return(stringers_final)

def web_frame_function(ice_class,min_power,displacement,web_lst, framing_sys ,
    T_bar_lst, selected_shell_plates_lst, effective_breadth, sigma_y):
    structure_type = "web"
    number_of_web_frames = len(web_lst)
    webs_final =[]
    for web in web_lst:
        web_index = web[0]
        web_x_cord = web[1]
        calculated_web = [[web_index,web_x_cord]]
        web.pop(0)
        web.pop(0)

        for web_part in web:
            if(web_part[0] == 1.0):
                location = "bow"
            elif(web_part[0] == 2.0):
                location = "mid"
            else:
                location = "aft"

            if(web_part[1] == 1.0):
                ice_reinforced = True
            elif(web_part[1] == 0.0):
                ice_reinforced = False

            if(web_part[2] == 1 or web_part[3] == 0):
                plate_thickness = selected_shell_plates_lst[0]
            elif(web_part[3] == 1):
```

```
                plate_thickness = selected_shell_plates_lst[1]
            elif(web_part[3] == 2):
                plate_thickness = selected_shell_plates_lst[2]
            elif(web_part[3] == 3):
                plate_thickness = selected_shell_plates_lst[3]
            elif(web_part[3] == 4):
                plate_thickness = selected_shell_plates_lst[4]
            elif(web_part[3] == 5):
                plate_thickness = selected_shell_plates_lst[5]
            elif(web_part[3] == 6):
                plate_thickness = selected_shell_plates_lst[6]
            elif(web_part[3] == 7):
                plate_thickness = selected_shell_plates_lst[7]
            elif(web_part[3] == 8):
                plate_thickness = selected_shell_plates_lst[8]
            elif(web_part[3] == 9):
                plate_thickness = selected_shell_plates_lst[9]
            elif(web_part[3] == 10):
                plate_thickness = selected_shell_plates_lst[10]
            elif(web_part[3] == 11):
                plate_thickness = selected_shell_plates_lst[11]
            elif(web_part[3] == 12):
                plate_thickness = selected_shell_plates_lst[12]

            true_web_span = web_part[4]
            web_span = web_part[5]
            web_spacing = web_part[6]
            profile, weight = web_requirements(ice_class,displacement,min_power,
    location,framing_sys,web_span,web_spacing,effective_breadth,plate_thickness,
    T_bar_lst,sigma_y)
            mass = weight*true_web_span
            print(profile)
            print(weight)
            calculated_part = [profile,weight,true_web_span,mass]
            calculated_web.append(calculated_part)
        webs_final.append(calculated_web)
    return(webs_final)

def ice_pressure(ice_class,displacement,min_power,location,structure_type,
    framing_sys,span,spacing):
    k = math.sqrt(displacement*min_power)/1000

    if (k <= 12 and location == "bow"):
        a = 30
        b = 230
    elif (k > 12 and location == "bow"):
        a = 6
        b =518
    elif (k <= 12 and (location == "mid" or location == "aft")):
        a = 8
        b = 214
    elif (k > 12 and (location == "mid" or location == "aft")):
        a = 2
        b = 286

    if ((a*k+b)/1000 > 1):
        c_d = 1
    else:
        c_d = (a*k+b)/1000

    if (ice_class == 1 and location == "bow"):
        c_p = 1.0
    elif (ice_class == 1 and location == "mid"):
        c_p = 1.0
    elif (ice_class == 1 and location == "aft"):
        c_p = 0.75
    elif (ice_class == 2 and location == "bow"):
        c_p = 1.0
```

```
    elif (ice_class == 2 and location == "mid"):
        c_p = 0.85
    elif (ice_class == 2 and location == "aft"):
        c_p = 0.65
    elif (ice_class == 3 and location == "bow"):
        c_p = 1.0
    elif (ice_class == 3 and location == "mid"):
        c_p = 0.70
    elif (ice_class == 3 and location == "aft"):
        c_p = 0.45
    elif (ice_class == 4 and location == "bow"):
        c_p = 1.0
    elif (ice_class == 4 and location == "mid"):
        c_p = 0.50
    elif (ice_class == 4 and location == "aft"):
        c_p = 0.25

    if (structure_type == "shell" and framing_sys == "transverse"):
        l_a = spacing
    elif (structure_type == "shell" and framing_sys == "longitudinal"):
        l_a = 1.7 * spacing
    elif (structure_type == "frame" and framing_sys == "transverse"):
        l_a = spacing
    elif (structure_type == "frame" and framing_sys == "longitudinal"):
        l_a = span
    elif (structure_type == "stringer"):
        l_a = span
    elif (structure_type == "web"):
        l_a = 2*spacing

    if (math.sqrt(0.6/l_a) > 1):
        c_a = 1
    elif (math.sqrt(0.6/l_a) < 0.35):
        c_a = 0.35
    else:
        c_a = math.sqrt(0.6/l_a)

    p0 = 5.6
    p = c_d*c_p*c_a*p0

    return(p)

def height_of_ice_load(ice_class):
    if (ice_class == 1):
        h = 0.35
    elif (ice_class == 2):
        h = 0.30
    elif (ice_class == 3):
        h = 0.25
    elif (ice_class == 4):
        h = 0.22
    return(h)

def shell_plate_thickness(ice_class,displacement,min_power,location,framing_sys,
    frame_span,frame_spacing,shell_plates_lst,sigma_y):

    t_c = 2
    p = ice_pressure(ice_class,displacement,min_power,location,"shell",framing_sys,
    frame_span,frame_spacing)
    ppl = 0.75*p
    h = height_of_ice_load(ice_class)
    if (1.3 - (4.2/((h/frame_spacing)+1.8)**2) > 1):
        f_1 = 1.0
    else:
        f_1 = 1.3 - (4.2/((h/frame_spacing)+1.8)**2)

    if (h/frame_spacing <= 1):
        f_2 = 0.6+(0.4/(h/frame_spacing))
```

```python
    else:
        f_2 = 1.4-0.4*(h/frame_spacing)

    if (framing_sys == "transverse"):
        shell_t_requirement = 667*frame_spacing*math.sqrt(f_1*ppl/sigma_y)+t_c
    elif (framing_sys == "longitudinal"):
        shell_t_requirement = 667*frame_spacing*math.sqrt(p/(f_2*sigma_y))+t_c

    for plate in shell_plates_lst:
        if plate >= shell_t_requirement:
            return (plate)

    return(shell_t)


def frame_requirements(ice_class,displacement,min_power,location,framing_sys,frame_span,frame_spacing,sigma_y):
    p = ice_pressure(ice_class,displacement,min_power,location,"frame",framing_sys,frame_span,frame_spacing)
    h = height_of_ice_load(ice_class)
    if (framing_sys == "transverse"):
        m_t = (7*7)/(7-5*h/frame_span)
        f_3 = 1.2
        Z = ((p*frame_spacing*h*frame_span)/(m_t*sigma_y))*10**6
        A = ((math.sqrt(3)*f_3*p*h*frame_spacing)/(2*sigma_y))*10**4
    elif(framing_sys =="longitudinal"):
        f_4 = (1-0.2*h/frame_spacing)
        f_5 = 2.16
        m = 13.3
        Z = ((f_4*p*h*frame_span**2)/(m*sigma_y))*10**6
        A = ((math.sqrt(3)*f_4*f_5*p*h*frame_span)/(2*sigma_y))*10**4

    return(Z,A)

def stringer_requirements(ice_class,displacement,min_power,location,within_ib,framing_sys,stringer_span,stringer_spacing,distance_to_ib,sigma_y):
    p = ice_pressure(ice_class,displacement,min_power,location,"stringer",framing_sys,stringer_span,stringer_spacing)
    h = height_of_ice_load(ice_class)

    if (within_ib == True):
        p_h_value = max(0.15,p*h)
        m = 13.3
        if (framing_sys == "transverse"):
            f_6 = 0.9
        else:
            f_6 = 1.0
        f_7 = 1.8
        f_8 = 1.2
        Z = ((f_6*f_7*p_h_value*stringer_span**2)/(m*sigma_y))*10**6
        A = ((math.sqrt(3)*f_6*f_7*f_8*p_h_value*stringer_span)/(2*sigma_y))*10**4
    elif (within_ib == False):
        p_h_value = max(0.15,p*h)
        m = 13.3
        if (framing_sys == "transverse"):
            f_9 = 0.9
        else:
            f_9 = 1.0
        f_10 = 1.8
        f_11 = 1.2
        Z =((f_9*f_10*p_h_value*stringer_span**2)/(m*sigma_y))*(1-distance_to_ib/stringer_spacing)*10**6
        A =((math.sqrt(3)*f_9*f_10*f_11*p_h_value*stringer_span)/(2*sigma_y))*(1-distance_to_ib/stringer_spacing)*10**4

    return(Z,A)
```

```
def web_requirements(ice_class,displacement,min_power,location,framing_sys,web_span
    ,web_spacing,effective_breadth,shell_thickness,T_bar_lst,sigma_y):
    p = ice_pressure(ice_class,displacement,min_power,location,"web",framing_sys,
    web_span,web_spacing)
    h = height_of_ice_load(ice_class)
    p_h_value = max(0.15,p*h)
    f_12 = 1.8
    f_13 = 1.1
    F = f_12*p*h*web_spacing
    Q = F
    M = 0.193*F*web_span
    i = 0
    for T in T_bar_lst:
        h_w = T[2]-T[4]
        C = 805
        web_t1 = h_w*math.sqrt(sigma_y)/C
        web_t2 = shell_thickness / 2
        web_t3 = 9
        web_t_requirement = max(web_t1,web_t2,web_t3)

        A_plate = (shell_thickness*effective_breadth*1000)/100
        I_x_plate = ((effective_breadth*1000*shell_thickness**3)/12)/10000
        y_x_plate = (shell_thickness / 2)/10

        A_profile = T[6]
        I_x_profile = T[8]
        y_x_profile = (T[11] + shell_thickness)/10

        y_c = ((A_plate * y_x_plate)+(A_profile * y_x_profile))/(A_plate +
    A_profile)

        H_plate = (y_c - y_x_plate)
        H_profile = (y_x_profile - y_c)

        I_c = (I_x_profile + A_profile*H_profile**2)+(I_x_plate + A_plate*H_plate
    **2)
        z = ((shell_thickness + T[2])/10 - y_c)
        Z = I_c / z

        A_f = T[1]*T[4]
        A_w = shell_thickness*effective_breadth
        A_a = A_f + A_w
        Af_Aw_lst = [0,0.2,0.4,0.6,0.8,1.0,1.2,1.4,1.6,1.8,2.0]
        alpha_lst = [1.5,1.23,1.16,1.11,1.09,1.07,1.06,1.05,1.05,1.04,1.04]
        gamma_lst = [0,0.44,0.62,0.71,0.76,0.80,0.83,0.85,0.87,0.88,0.89]
        u = 0
        while (A_f/A_w < Af_Aw_lst[u]):
            u+=1

        if (A_f/A_w >= 2.0):
            alpha = alpha_lst[-1]
            gamma = gamma_lst[-1]
        else:
            alpha = alpha_lst[u-1]+(A_f/A_w - Af_Aw_lst[u-1])*((alpha_lst[u]-
    alpha_lst[u-1])/(Af_Aw_lst[u]-Af_Aw_lst[u-1]))
            gamma = gamma_lst[u-1]+(A_f/A_w - Af_Aw_lst[u-1])*((gamma_lst[u]-
    gamma_lst[u-1])/(Af_Aw_lst[u]-Af_Aw_lst[u-1]))

        A_requirement = ((math.sqrt(3)*alpha*f_13*Q)/(sigma_y))*10**4
        Z_requirement = (M/sigma_y)*math.sqrt(1/(1-(gamma*A_requirement/A_a)**2))
    *10**6

        if (web_t_requirement <= T[3] and A_requirement <= A_profile and
    Z_requirement <= Z):
            return(T[0],T[7])

    return("No Profile",0)

```

```

def bulb_profile_selection(shell_thickness,effective_breadth,Z_requirement,A_requirement,bulb_list,T_bar_lst,sigma_y):

    i = 0
    for bl in bulb_list:
        h_w = bl[1]-bl[4]
        C = 805

        web_t1 = h_w*math.sqrt(sigma_y)/C
        web_t2 = shell_thickness / 2
        web_t3 = 9  #mm
        web_t_requirement = max(web_t1,web_t2,web_t3)

        A_plate = (shell_thickness*effective_breadth*1000)/100
        I_x_plate = ((effective_breadth*1000*shell_thickness**3)/12)/10000
        y_x_plate = (shell_thickness / 2)/10

        A_profile = bl[6]
        I_x_profile = bl[10]
        y_x_profile = (bl[8] + shell_thickness)/10

        y_c = ((A_plate * y_x_plate)+(A_profile * y_x_profile))/(A_plate + A_profile)

        H_plate = (y_c - y_x_plate)
        H_profile = (y_x_profile - y_c)

        I_c = (I_x_profile + A_profile*H_profile**2)+(I_x_plate + A_plate*H_plate**2)
        z = ((shell_thickness + bl[1])/10 - y_c)
        Z = I_c / z

        if (web_t_requirement <= bl[2] and A_requirement <= A_profile and Z_requirement <= Z):
            return(bl[0],bl[7])

    profile, weight = T_bar_profile_selection(shell_thickness,effective_breadth,Z_requirement,A_requirement,T_bar_lst,sigma_y)

    return(profile,weight)


def T_bar_profile_selection(shell_thickness,effective_breadth,Z_requirement,A_requirement,T_bar_lst,sigma_y):
    i = 0
    for T in T_bar_lst:
        h_w = T[2]-T[4]
        C = 805

        web_t1 = h_w*math.sqrt(sigma_y)/C
        web_t2 = shell_thickness / 2
        web_t3 = 9
        web_t_requirement = max(web_t1,web_t2,web_t3)

        A_plate = (shell_thickness*effective_breadth*1000)/100
        I_x_plate = ((effective_breadth*1000*shell_thickness**3)/12)/10000
        y_x_plate = (shell_thickness / 2)/10

        A_profile = T[6]
        I_x_profile = T[8]
        y_x_profile = (T[11] + shell_thickness)/10

        y_c = ((A_plate * y_x_plate)+(A_profile * y_x_profile))/(A_plate + A_profile)

        H_plate = (y_c - y_x_plate)
        H_profile = (y_x_profile - y_c)
```

```

        I_c = (I_x_profile + A_profile*H_profile**2)+(I_x_plate + A_plate*H_plate**2)
        z = ((shell_thickness + T[2])/10 - y_c)
        Z = I_c / z

        if (web_t_requirement <= T[3] and A_requirement <= A_profile and Z_requirement <= Z):
            return(T[0],T[7])

    return("No Profile",0)



if __name__ == '__main__':
    main()
```

NTNU
Norwegian University of
Science and Technology

ELOMATIC
CONSULTING & ENGINEERING