Sabyasachi Ghosh Dastidar

Capacity of Ice-Strengthened Structures subjected to Different Ice-loading Scenarios

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 30.1.2017

Thesis supervisor:

Professor Jani Romanoff

Thesis advisors:

Mihkel Körgesaar D.Sc (Tech.)

Mikko Suominen D.Sc (Tech.)

Aalto University

Aalto University
School of Engineering

Aalto University, P.O. BOX 11000, 00076 AALTO

www.aalto.fi

Abstract of master's thesis

Author	Sabyasachi Ghosh Dastidar
Title of thesis	Capacity of Ice-Strengthened Structures subjected to Different Ice-loading Scenarios
Degree programme	Master's degree programme in Mechanical Engineering
Major/minor	Innovative structural design (Minors: Mechanics of Materials and Product Development)
Code	IA3027
Thesis supervisor	Professor Jani Romanoff
Thesis advisor(s)	Mihkel Körgesaar D.Sc (Tech.) and Mikko Suominen D.Sc (Tech.)
Date	30.1.2017
Number of pages	55+28
Language	English

Abstract

This thesis investigates the critical loading scenarios of multiple ice-strengthened ship structures when subjected to different load configurations. Evaluation of structural performance has been conducted using Non-linear Finite Element Analyses. Using a non-uniform pressure patch (NUPP), the capacity of ship structures was investigated and the safety margins against different capacity levels was determined. Results show that the point of yield and load for permanent deformation are highest for a structure built with IA Super ice class of the Finnish-Swedish Ice Class Rules (FSICR). The values are observed to decrease for weaker ice classes. Although the present approach of FSICR is to design structures to yield once per winter, the safety factor against yield was about two for ice classes IA and IB, but slightly lower for IA Super.

Comparison of different loading approaches, namely present design approach and non-uniform pressure patch, show that the FSICR approach yields much stiffer response. This justifies the use of less conservative NUPP for structural capacity evaluation, results of which show that the present FSICR design approach is extremely conservative. Alternatively, the current design pressure patch approach renders structures more safe than they actually are. Therefore, future investigations must clarify the pressure profile due to the ice loads.

Keywords Winter navigation, Finnish-Swedish Ice Class Rules, ice-loading scenarios, limit states, ice-strengthened structures, decreasing load function.

Foreword

This master thesis was carried out at the *Marine Technology* research group at *Aalto University School of Engineering* in Espoo, Finland in cooperation with *TRAFI* (Finnish Transport and Safety Agency). The case studies conducted in this thesis were focused on the domain of capacity of ice-strengthened ships when multiple ice-loading scenarios are applied on different locations. For this, three ice classes from Finnish Swedish Ice Class Rules have been used. The thesis research was carried out over a period of 6 months from mid-May to December.

I would like to thank my instructor *Mihkel Körgesaar* for his supreme guidance throughout my thesis. Without his in-depth expertise and timely help, this thesis would lack proper results. I would also like to thank my other instructor *Mikko Suominen* for his critical help on important areas required to steer the direction of this work. I am grateful to *Jani Romanoff* for his regular comments on thesis drafts and creating schedules which kept this thesis on track. Thanks also to *Pentti Kujala*, the project coordinator, and *Jorma Kämäräinen*, the TRAFI representative, for their feedback on the monthly progress presentations.

Last but not least, I would like to thank the other colleagues at Marine Technology for troubleshooting some other technical issues during my thesis and my family for supporting me throughout my academic journey.

Espoo, January 30th 2017

Sabyasachi Ghosh Dastidar

Table of contents

List of Symbols

p_{av}	Uniform ice pressure
h	Ice load height (rectangular load patch)
h_o	Level ice thickness
l, l_a, l_c	Load length
p_o	Nominal ice pressure
q	Line load
s	Frame spacing
c_d	Ship size and engine output factor
c_p	Hull region factor
c_a	Coefficient dependent on the load length
C, a	Unknown parameters specific to each ship
IAS	IA Super ice class
IA	IA ice class
IB	IB ice class

1 Introduction

The Baltic Sea has become one of the major maritime highways, with winter traffic accounting for one-quarter of the annual traffic. In 2007, 850 million tons of cargo was handled in Baltic ports. In addition, owing to its excellent network, the Baltic passenger transportation sector conveys over 30 million people annually.

Winter navigation has been an important factor allowing continuous maritime navigation in the Baltic, which makes it a key driver for Finnish economy. By effectively utilizing the winter navigation system, maritime incidents can be significantly reduced. This system is based on icebreakers, ice-strengthened merchant fleet and ice classes such as the Finnish-Swedish Ice Class Rules (FSICR).

Demand for ice-strengthened ships built in accordance with design standards such as FSICR have steadily increased to ensure effective winter navigation (Riska et al., 2011). The main intention of FSICR has been to balance safety with commercial benefits. A FSICR ice class has been shown to have a direct impact on the susceptibility of ship to ice damage (Kujala, 1991 and Hänninen, 2003). The probability of ice damage is nine times greater for ships of a lower ice class. For structural design purposes, the ice load is assumed in FSICR to be uniform ice pressure on a rectangular load patch. Although the current design point of FSICR is to allow the yield limit to be reached at least once per winter, ice loads can also permanently damage the ship structure. This damage can lead the ship to its *limit state,* at which point a structure or a structural member becomes unfit for one of its intended roles.

Although FSICR is the most commonly applied design standard for ice-strengthened ships in the Baltic waters, most of the assumptions made in the rules are based on empirical experience (TRAFI, 2010). These empirical experiences from the operating ships have led to the current state of the structural design through measurements and damage surveys (Kujala, 1991). Naturally, there is a high level of uncertainty associated with design ice loads in FSICR. The design loads lead to a ship structure that has a certain load carrying capacity. This is in line with the intention for the rules, which is to define an acceptable amount of damage to the ship structure that would still allow the transportation to be economical but also safe. However, the actual load carrying capacity has

not been clearly defined in the rules. The reason for this is that FSICR does not explicitly specify acceptable permanent deformation or the safety margin of shell plating and framing.

To address these gaps in the rules, this thesis evaluates the structural performance of ship structures using non-linear Finite Element (FEM) Analyses for the purpose of determining the actual load-carrying capacity. In order to accomplish this, fifteen case studies have been used to analyse the response of different FEM models of several locations on the ship hull when subjected to various ice-loading scenarios. Each of the FEM models of the hull locations has been constructed with scantlings obtained using three ice classes: IA Super, IA and IB. In addition, this thesis determines the capacity levels and safety margins of the side structures using a non-uniform pressure patch. Among the different limit states, this thesis focuses on the *serviceability limit state* (SLS), which denotes structural damage while the structure still keeps its primary load-carrying role (Hughes and Paik, 2010). Specifically, the analysis will focus on permanent deformations in the side structure due to ice loads.

The rest of this thesis is divided into five chapters. Chapter 2 is devoted to discussing the state-of-the-art for Finnish-Swedish Ice Class Rules, capacity of ice-strengthened structures and ice load definitions. Chapter 3 is used to clearly define the scope of this work. Chapter 4 explores the theoretical background of the Finite Element descriptions. Chapter 5 describes the methods, in which the approaches used for modelling ice loads have been discussed. Chapter 6 documents the details of ships used for the case studies as well as explains the finite element models. Chapters 7 and 8 are devoted to results and discussion and defining the final conclusions of this work respectively.

2 State of the art

This thesis is mainly focused on Finnish-Swedish Ice Class Rules, capacity of ice-strengthened structures and different ice load definitions. In this chapter, a broad overview is provided for these topics.

2.1 Finnish-Swedish Ice Class Rules

Ice-strengthened ships navigating in the Baltic waters are built in accordance with design standards such as Finnish-Swedish Ice Class Rules (FSICR). These rules have become an integral part of the winter navigation system enabling a balance between commercial benefits and safety of marine traffic. FSICR includes requirements for the ship hull, machinery and ship performance in ice. If ships have low ice performance, it would need the help of icebreaker escorts. The structure of the rules is shown in Figure 2.1.

There are four ice classes according to the FSICR, which are IA Super, IA, IB, and 1C (IA Super being the highest order of strength). The strength level in by each of these ice classes roughly corresponds to the loading resulting from a certain level of ice thickness. For IA super, nominal ice thickness is 1m (highest) whereas for 1C, it is 0.4m (lowest). The IA super, being of the highest order, is defined such that its ice thickness is higher than the maximum ice thickness level observed in the Baltic outside the fast ice zone (Riska *et al.,* 2011).

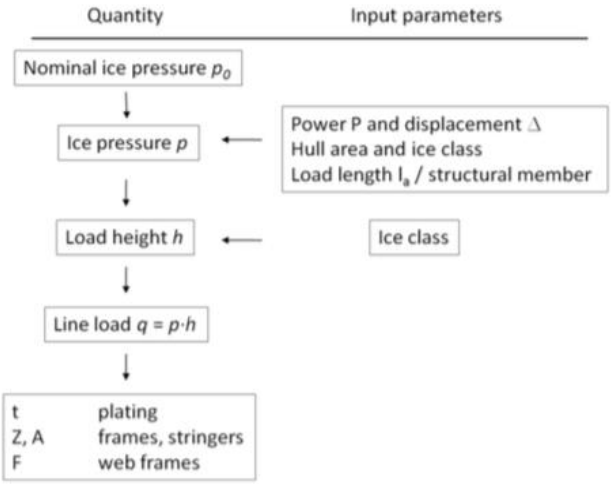

Figure 2.1. Structure of the FSICR (Maritime Safety Regulation, 2010)

2.1.1 Ice loading

As the ice load is a statistical quantity, the design load needs to be determined by an assumption of probability level or return period of the load (Riska *et* al, 2011). However, the load level defined in the present FSICR is determined based on the experience of ice-induced hull damage. The loading should be viewed together with the allowed response (yield point or plastic deformation). This enables the loading probability and the expected response to be in balance.

Ice load definition is a significant part of the hull rules. For the purpose of structural design, the standard procedure is an assumption that ice load can be described by uniform ice pressure, termed p_{av}, on a rectangular load patch of length l_a and height h. This approach of constant pressure is taken as the material properties of the Baltic ice do not change much through the winter in different Baltic Sea areas. Therefore, the total force is $F = p_{av} \cdot h \cdot l_a$, (Riska *et al.,* 2011).

The total ice load for each structural member is taken as the line load q times the load length l_a that depends on the distance between respective structural members (horizontal span or spacing). For transverse frames the load is, for example, $F = q \cdot s$, where s is the frame spacing.

The direct analysis of ship hull structures can be carried out using a load patch with p_{av}, h and l_a defined. The pressure to be used is $1.8p_{av}$, where design ice pressure p_{av} is calculated with equation 1 (Maritime Safety Regulation, 2010). The application of this load patch is at locations where the capacity of the structure is minimized under the combined effects of shear and bending. The analysis of results of SAFEICE project highlighted the fact that the design loads in the FSICR are relatively low – but in balance with the design point which is first yield.

The design ice pressure is calculated by:

$$p_{av} = c_d \, . c_p \, . c_a \, . p_0 \text{ MPa} \quad \text{(1)}$$

where

c_d is a factor accounting for influence of the ship's size and engine output

c_p is a factor accounting for the probability that the design ice pressure is being applied at a certain location of the ship hull (ice-class specific)

c_a is a factor accounting for the probability that the entire length of the patch area will be subjected to the pressure at a particular instance in time

p_o is nominal ice pressure with value 5.6 MPa

There are various approaches to computationally model ice loads due to the stochastic nature of ice failure processes during ice-structure interactions. The design practice for designing and analyzing ice-strengthened ship structures is the assumption of the stationary, uniform pressure load (Maritime Safety Regulation, 2010) which is exerted upon patches on framing or plating. However, recent research on this practice (Quinton *et al.,* 2012 *and* Erceg *et al.*, 2014) has highlighted that such pressure loads might lead to conservative results in comparison to actual measured field data, or in comparison to more complex load shapes and moving loads (Kõrgesaar and Kujala, 2016). Another method to proceed would be to represent the ice-induced load to be decreasing as a function of load length, which is discussed in Chapter 2.3.

2.1.2 Influence of ice load height

Regarding ice load height, there is an assumption that the operation of an ice-strengthened ship is limited to open sea conditions that corresponds to ice thickness h_o as shown in Table 2.1 (TRAFI, 2010). Another assumption is that the corresponding design ice load height h of the area subjected to ice pressure is only a segment of the ice thickness.

Table 2.1. Load height for each ice class

Ice Class	h_o [m]	h [m]
IA Super	1.0	0.35
IA	0.8	0.30
IB	0.6	0.25
IC	0.4	0.22

2.1.3. Influence of hull area and ice load length

In all ice class rules, the hull loading depends on the location of the hull where the loading is defined. The bow, mid body and stern encounter different loads. As the loads on different areas are of different origin, it is also evident that the loading magnitude is different. This is taken into account in most of the ice class rules by dividing the ship hull into areas and defining hull factors c_p for each area. These hull area factors relate the loading of that particular area to that of the bow.

In the FSICR three ship hull areas are used: bow, midbody and stern. The loading is also assumed to be acting on narrow area in vertical direction. The strengthened area is called the ice belt. Each of the three hull regions has a design ice pressure defined by a hull region factor c_p. This factor accounts for the probability that the design ice pressure will be occurring at a certain region of the ship hull for each specific ice class. c_p is defined initially for the bow region, and is scaled

according to the ice class for other regions so that the stern region has the lowest design ice pressure. For ice class IC, the stern hull region coefficient is 0.25 (as shown in Table 2.2).

Table 2.2. Hull region factor c_p for different locations on the ship hull

Ice Class	Region		
	Bow	Midship	Stern
IA Super	1.0	1.0	0.75
IA	1.0	0.85	0.65
IB	1.0	0.70	0.45
IC	1.0	0.50	0.25

Another factor used to define the ice pressure is a coefficient dependent on the load length, c_a. Each structural member has an associated load length l_a – this is the length of the load that influences the response in the hull area. The maximum line load can also be represented as a function of the loading length, with the magnitude of ice-induced line load decreasing as a function of load length (Kujala and Arughadhoss, 2012).

2.2 Capacity of ice-strengthened structures

According to Finnish Transport and Safety Agency (TRAFI), web frames and load-carrying stringers are to be dimensioned such that the combined stresses from shear and bending, using the von Mises yield criterion, are lower than the yield point. In other words, the intention is to design structures to remain always in the elastic region so that no permanent deformations due to plasticity take place. When steel material, commonly used in ship structures, becomes plastic it loses stiffness – deformations due to loads increase faster than they do in the elastic regime. These deformations are permanent meaning that once the load is removed, deformations remain, see Figure 2.2. The loss of structural stiffness lends itself well to the definition of structural capacity.

Commonly, as in Figure 2.2, capacity is associated with reaching a certain permanent deformation in the structure. The safety of the structures is thus directly linked to the structural capacity, which however is vaguely defined for ice strengthened ship structures exposed to ice loads.

Two different capacity measures are often associated with ice loads:

1) **Serviceability limit state (SLS):** The permissible failure limits for damage in normal conditions are defined by this limit state. The scope of this limit state is for local failures affecting the structural durability, its aesthetic appearance, and which doesn't result in its

ultimate failure. These local failures can include deflections in hull plating of ships which lead to rise in fuel costs due to an increase in resistance. However, the damage doesn't have an effect on the overall strength of the ship hull. Serviceability limit state according to DNV is permanent deformation of $s/12$, where s is the frame spacing.

2) **Three-hinge mechanism:** In this, the deformation rises infinitely with a partial increase in load-level. Because of the hypothetical nature of the three-hinge mechanism, methods such as 0.1% residual strain method (Figure 1.2) is needed to determine the limit load value. In this method, the capacity is equal to the load causing permanent strain of 0.1% span (Abraham, 2009).

While clearly defined, these measures do not fully clarify the additional reserve capacity ice strengthened ship structures have, which is important when considering more critical loading scenarios. Hence, the objective of this work is to investigate the capacity of ice strengthened ship structures in more detail and compare the response with current design criteria.

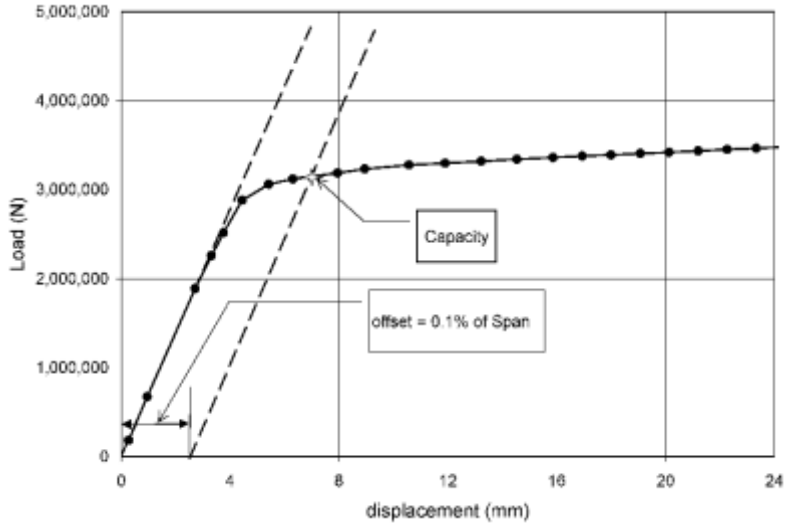

Figure 2.2. Capacity of a structure defined using 0.1% residual strain method (Abraham, 2009)

2.3 Ice load definitions

2.3.1 Decreasing line load as a function of load length

The function relating measured maximum line load with loading length has been researched on by Kujala *et al.* (2012) and Riska *et al.* (2011). The function, defined in equation 2, is fitted to multiple

ship measurement data to render curves denoting the magnitude of line load q as a function of load length.

$$q = C\left(\frac{lc}{s}\right)^{-a} \qquad \text{..... (2)}$$

where s is the frame spacing

l_c is the load length

C and a are parameters specific to each ship.

The study included one data set (Antarctic data) divided into one and twelve-hour time periods. The line load maxima were calculated for each time period for the bow and stern. In each location of the ship hull, multiple load lengths were considered for these calculations which corresponded to the number of frame spacing. C and a are determined with non-linear least square method by curve-fitting of equation 2 with each ship's measurement data (Figure 2.3). For the case of ship S.A. Agulhas II, s is 0.4m, the C is obtained as 1989 kN/m and a is 0.733 for the stern and 3150 kN/m and 0.538 for the bow. It has to be noted that the results differ whether 1 or 12 hour maxima is employed when the return period of time is determined. It is also to be noted that the exponent value a in equation 2 was selected to be 0.5 for the rules (Riska and Kämäräinen, 2011).

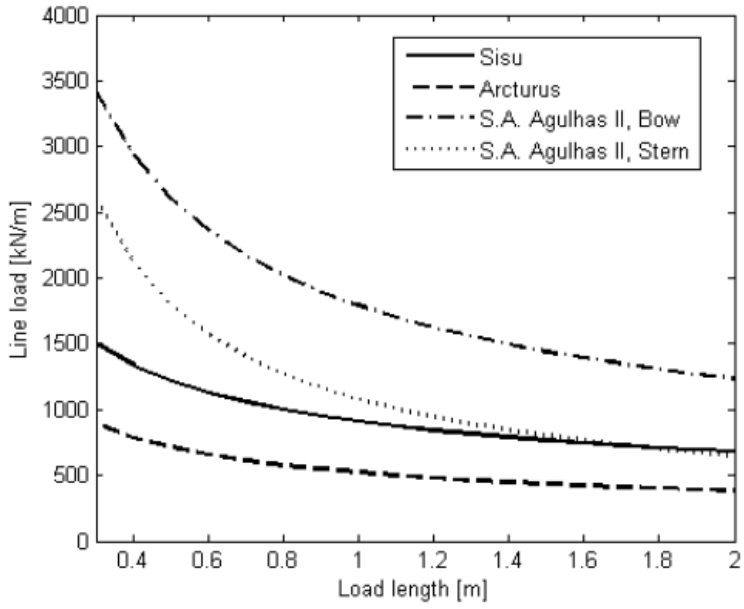

Figure 2.3. Maximum line loads determined from full-scale ship measurements (three ships) as a function of load length (Suominen *et al.*, 2015). (Return period taken is 10 days)

2.3.2 Ice-load sharing in ship structural design

When ice loading is applied in ship structure design, frames are considered individually. However, when this applied loading is non-uniform or non-symmetric (Figure 2.4), the most heavily loaded frame distributes some of its load with its neighboring frames (Abraham, 2009).

Abraham's work examined the difference in the capacity of single frames in isolation compared to single frames as part of a grillage. This is a critical aspect in the design of ice class framing as the boundary conditions influence the stiffener response. Load-deflection characteristics of the center stiffener in a large grillage was compared to that of a single frame model. Both models were subjected to asymmetric loading during the comparison. Results of this research on load sharing behavior has shown that there is a variance in the behavior of a frame when it is in a grillage. In addition, it was observed that the higher ice class structures have lower capability of load distribution. Hence, according to the proposed theory by Abraham, higher ice class structures will be subjected to higher as well as more local ice loads.

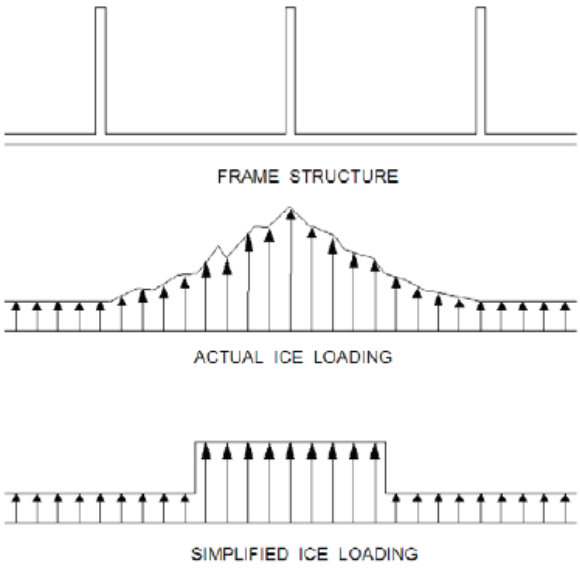

Figure 2.4. Non-uniform loading and uniform ice loading on frames (Daley and Kendrick, 2008)

Other notable conclusions are that large grillage also results in the structure being 0-36% stronger than the single frames. This increased strength equalizes the uncertainties in ice load estimation, and hence acts as a safety margin for the structure. It has also been shown that the thinner frames deform together as a structure (more load sharing) whereas thicker frames show local deformation

pattern. Therefore, the single frames become stronger when they are part of a grillage as they can share their load with adjacent frames.

2.3.3 Other approaches used to model ice loads

From recent research (Quinton *et al.,* 2012; Erceg *et al.*, 2014), it has been concluded that there is a high level of uncertainty in assessing the risk level due to the stochastic nature of ice failure processes during ice-structure interactions. For instance, Quinton *et al.* (2012) and Kõrgesaar *et al.* (2016) showed that structural response is strongly affected by the load application method, i.e. whether exerted load is moving or decreasing in height during application process. This often yields more structural damage than the equivalent pressure patches for stationary, uniform pressure loads that are compliant with Classification Society rules.

2.3.4 Design loads defined in IACS in contrast to FSICR

The derivation of the design ice load for the rules concerning polar class (IACS) is based on the scenario of a glancing collision between the ship's bowshoulder and ice edge (Lyngra, 2014). The derivation of loading is based on the ship's kinetic energy balancing the energy of crushing ice, as shown in equation 3. The ice thickness, ship size & speed, hull form and ice strength are also used in the derivation.

$$\frac{1}{2} M_e \cdot V_n^2 = \int_0^{\delta} F_n(\delta) \cdot d\delta \qquad \text{..... (3)}$$

In the above equation, the left hand side denotes the ship's kinetic energy (in terms of variables ship mass and velocity), whereas the right hand side gives the energy used for crushing the ice (in terms of integration of the normal force over the penetration depth).

In contrast to IACS rules, the determination of ice loads stated in FSICR have been based on experience of ice-induced hull damage of marine winter traffic at the Baltic Sea. According to FSICR design ice load rules, the loads have been considered as uniform ice pressure and calculated using equation 1.

3 Numerical Simulation Methods

During the last few years, finite element analysis has become important in the domain of ship engineering. In this thesis, it has also been intensively used to achieve the goals of this work. In this chapter, a broad overview of the Finite Element Analysis is introduced with a discussion of the FEM descriptions. Moreover, general aspects of FEM analysis have been presented such as shell elements, geometric and material nonlinearities, strain hardening as well as implicit/explicit analysis. The specific details of FEM models used for obtaining the results of this thesis have been discussed in Chapter 6.2.

3.1 Introduction to Finite Element Analysis

In Finite Element Method, a large model is subdivided into smaller finite elements in order to obtain an approximate solution for a boundary value problem. For failure of large complex structures, the nature of failure is always considered to be nonlinear. A significant advantage of Nonlinear FEM is that it can consider geometric nonlinearities such as large deflections and rotations. In addition, material nonlinearities (such as yield, plastic deformation) and contact nonlinearities can be considered in FEM analyses. Finite Element solvers such as Abaqus, Ansys and Comsol are commonly used for structural analysis of ship hull. In these FE solvers, two-dimensional conventional shell elements can be used to model thin-walled structures. The loading and boundary conditions are applied on the reference surface of these shell elements, which is defined by its nodes and normal direction (Figure 3.1).

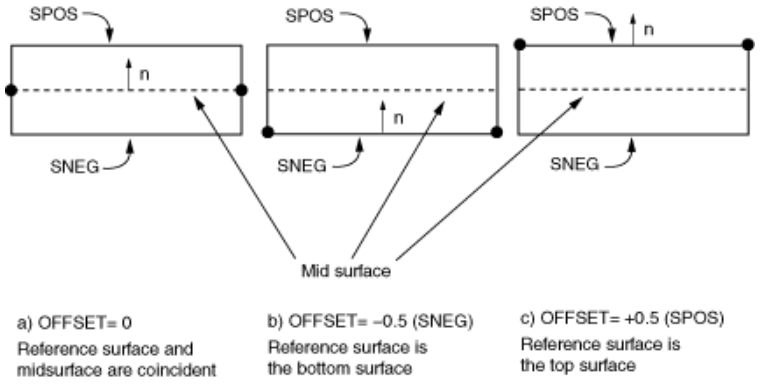

Figure 3.1. Shell offset: (a) 0: Mid-surface is taken as reference surface; (b) -0.5: Bottom surface (SNEG) is taken as reference surface; c) +0.5 Top surface (SPOS) is taken as reference surface.

Present design methods for ship hull structures use empirical data to assess the design pressure as a result of ice-structure interaction. On the other hand, finite element simulations for ice-structure interactions are a series of experiments in which simulation of nonlinear material and geometric behaviour can be performed. With such simulations, behaviour of ship hull and its material can be simulated on a large scale. There have been several recent developments in numerical simulations towards the construction of design pressures and ice-strengthened structures. For instance, the equivalency concept was introduced by Finnish Maritime Association (FMA) and application of nonlinear FEM was accepted as a viable substitute to ice class requirements (Wang *et al.,* 2006). This was followed by an initiative by American Bureau of Shipping (ABS) for development of formal procedures to use nonlinear FEM approach for designing alternative ice-strengthened structures (Wang *et al.,* 2005).

Using FEM approach, specifications such as extreme ice loads, structural and material modelling can be made, as shown by Wang. Based on this approach, FMA has formalized the new criteria for side longitudinal and side shell, which rendered the FSICR criteria for side stringer under debate. In the past decade, multiple research projects have been conducted in this domain. For instance, Wang has also conducted a study in which the ice loads and stringer thickness were varied and the impact on the ultimate and buckling strength of the side stringer was investigated.

During the last few decades, the usage of Finite Element Method has become an important tool in ship engineering and is preferred over analytical methods that cannot handle the geometric and material nonlinearities involved in analysing ice-structure interactions. However, the complexities involved in these simulations (due to parameters such as boundary conditions, loading, mesh density and type of modelling) can translate to discrepancies between the applied simulation methods and actual experimental results. These discrepancies most commonly arise due to the inadequacy of the modelling parameters to idealise the actual structure. Hence, the Finite Element model design has to be a compromise that reflects a high level of accuracy, and at the same time balances it with limited available modelling time.

3.2 Linear and Non-linear solutions to FEM problems

The Finite Element Method is based on element discretization of the concerned structure. Equilibrium equations are solved for each element in the structural model to obtain an approximate solution to its response. The equilibrium equations can be formulated based on the principle of virtual work (Reddy, 2014). The equations of virtual work are numerically solved in an element to provide an approximate solution for the equilibrium equation (equation 4).

$$\mathbf{K}\,\mathbf{U} = \mathbf{F} \quad \text{(4)}$$

The above equation represents a system of equilibrium equations over the element domain. **K** is the stiffness matrix of the element, which has been obtained by a summation of individual element stiffness matrices. The stiffness matrix of the structure is a function of its geometric and material properties. **U and F** are vectors which contain the nodal displacements and forces respectively.

For linear system of equilibrium equations, direct methods are used in which nodal displacements are solved by using Gauss elimination to manipulate the equations. Further, the nodal displacements can be used to calculate the strains and stresses. However, for nonlinear system of equations, such direct methods cannot be used to obtain the solution. In such cases, the finite element solution has to be obtained at each increment, and the final solution is developed incrementally. Most commonly, Non-linear Finite Element softwares use a static solver and the solution procedure is based on Newton's method, in which the nonlinear equation is solved by iteratively approximating the solution with Taylor series. In the static solver, quasi-static response of the structure is assumed because the time dependent mass and inertia effects are assumed to be minimal and are neglected. The Newton's method (also called the Newton-Raphson method) is shown graphically in Figure 3.2.

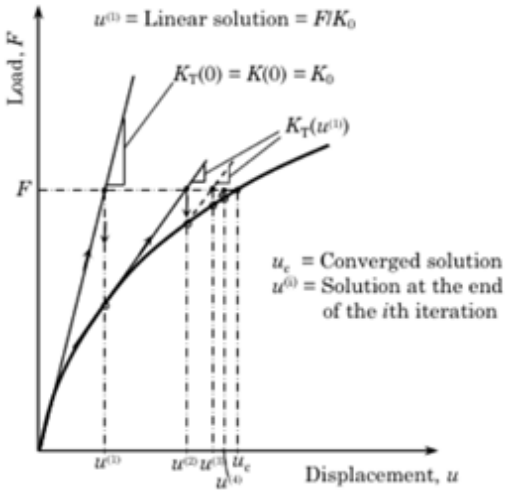

Figure 3.2. Solution procedure for Newton-Raphson method used for non-linear finite element problems. (Reddy, 2014).

In a nonlinear FEM problem, the stiffness matrix **K** depends on either the load vector or displacement vector. $^{1}\mathbf{U}^{0}$, which is an approximation of the solution $^{1}\mathbf{U}$, is calculated for the equilibrium equation at each step for the incremental force $^{1}\mathbf{F}$, as shown in equation 5.

$$^{1}\mathbf{K}^{0}\ {}^{1}\mathbf{U}^{0} = {}^{1}\mathbf{F} \quad \text{(5)}$$

In each iteration, an error is obtained with respect to force $^{1}\mathbf{F}$ that is continuously reduced until acceptable tolerance is achieved, at which point the solution is said to converge. This iteration procedure of Newton's Method is shown in equation 6.

$$\mathbf{K}^{i-1}\ \Delta\mathbf{U}^{i} = \mathbf{F} - \mathbf{R}^{i-1} \quad \text{(6)}$$

where

$$\Delta\mathbf{U}^{i} = \mathbf{U}^{i} - \mathbf{U}^{i-1} \quad \text{(7)}$$

The new estimate for **U** is obtained from the solution to the first iteration round, which is further used in the next iteration round until convergence. Typically, several iterations are required in a nonlinear problem to reach a converged solution. In each of the iterations rounds involved in obtaining the final solution, the stiffness matrix is recalculated and the nodal displacement **U** is newly estimated at force **F**. After the nodal displacements are obtained, the elemental strains and stresses can also be solved using interpolation matrices between displacement, stress and strain.

3.3 Shell elements in Finite Element Models

Element properties are important in Finite Element analysis in order to obtain accurate solutions to the equilibrium equations. The selection of elements should be done by considering the element's properties to realistically model material behaviour and deformations (degrees of freedom). Shell elements are commonly used to model large thin-walled structures such as ship hull. This is because shell elements neglect stresses in the thickness direction which saves valuable computation time in analysis of large thin-walled structures. Being in a plane stress state, shell elements are a good approximation for thin plates.

Conventional four-node shell elements have six degrees of freedom at every node, and discretise the represented volume by reference surface. The thickness of the elements can be defined using section properties in Finite Element solvers. Interpolation functions such as isoparametric mapping are used to interpolate the nodal displacements and rotations on the element reference surface. Isoparametric mapping lets the integrals in quadrilateral elements (*xyz*-coordinates) to be calculated in regular rectangular geometry ($\xi\eta$-coordinates), as shown in Figure 3.3.

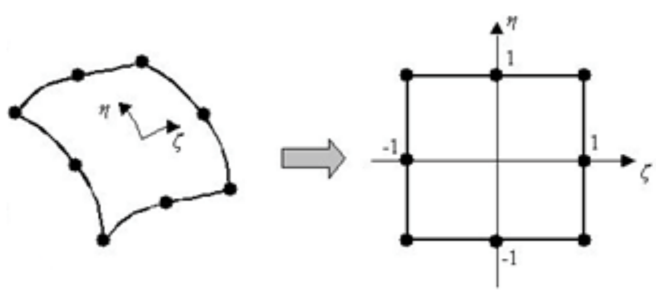

Figure 3.3. A four-node shell hell element as used in a ship hull finite element model. Physical quadrilateral elements (x, y, z coordinates) can be converted to regular rectangular geometry ($\xi\eta$-coordinates) using isoparametric mapping.

This type of interpolation is performed to improve the effectiveness of analysis due to the irregularity of shapes in the element space, which makes it simpler over regular geometry. For this transformation from element to regular geometry, numerical integration is applied (equation 8). Numerical integration is used for evaluation of stiffness matrices, mass matrices, surface loads etc.

$$\int \boldsymbol{P}(\xi,\eta)\, d\xi\, d\eta = \sum_{i,j} \alpha_{ij} \boldsymbol{P}(\xi_i,\eta_j) \quad \text{……………………..……..……...….} \ (8)$$

This equation shows the element matrix to be integrated with P, which is evaluated at the sampling points (ξi, ηj). αij are weight coefficients corresponding to these sampling points. The locations of sampling points as well as the weight coefficients can be determined using Gauss numerical integration (Bathe, 2016) or reduced integration. In this thesis, reduced integration has been preferred numerical integration for analysis as it improves the accuracy of results as well as reducing computational time. For reduced integration, the calculation of stresses and strains are done at points such that the overall solution is optimised for accuracy. For instance, integration is performed at the middle of the shell element in the case of four-node quadrilateral shell element. In the case of nonlinear analysis, several integration points are used to calculate section properties. These integrations points lie along the thickness of the shell element, while the change in thickness is a function of the membrane strain and Poisson's ratio.

3.4 Geometric and material non-linearity

The effects of geometric non-linearity is a result of the changes in geometry during analysis (large displacement analysis), and is characterised by the non-linear displacement-strain relation. During this type of non-linearity, stiffness matrix is constantly updated after each loading increment due to which the deformed geometry (updated matrix) is taken into consideration in the next analysis "step".

A general thumb rule is to use nonlinear geometry in the analysis when strains exceed 5%. In a Finite Element solver such as Abaqus, geometric non-linearity is taken into account by switching on the corresponding parameter *nlgeom*. This setting can be found in the Step manager, which defines the general non-linear analysis steps. This means that the state of the model at the end of one general step is the initial state of the model for the start of the next general step.

Material nonlinearity, on the other hand, is a result of the changes in material properties during analysis (plasticity, viscoplasticity etc.). This type of nonlinearity can be characterised by inelastic behaviour as seen in the non-linear stress-strain relationship. Once the structure reaches the yield point, further loading will cause response to deviate from its initial elastic behaviour. This nonlinear response, called strain hardening, may increase till it reaches an ultimate point before degrading (softening). Power law model is one of the various ways that can be used to model material non-linearity.

In the power law model, hardening curve is obtained by fitting the power law equation (equation 9) to the post-yielding part of the true stress-strain curve.

$$\sigma_f = K \varepsilon^n_p \quad \text{..... (9)}$$

Before the plastic zone, linear-elastic material behaviour is assumed with a specified elastic modulus.

3.5 Strain hardening

After the yield point is reached, hardening is said to occur and the stress needs to be continually increased to continue the plastic deformation. There is a set of hardening rules that define how the yield surface changes with plastic deformation. Two of the most commonly used hardening rules are isotropic and kinematic hardening.

In the case of isotropic hardening, the yield surface maintains its shape but is seen to expand as stress is increased (Figure 3.4). The yield is taken into account in the form of a function that is shown in equation 10.

$$f(\sigma_{ij}, K_i) = f_0(\sigma_{ij}) - K = 0 \quad \text{..... (10)}$$

The initial yield function denotes the shape of the yield function. The change in the size of this function is due to a change in the hardening parameter K.

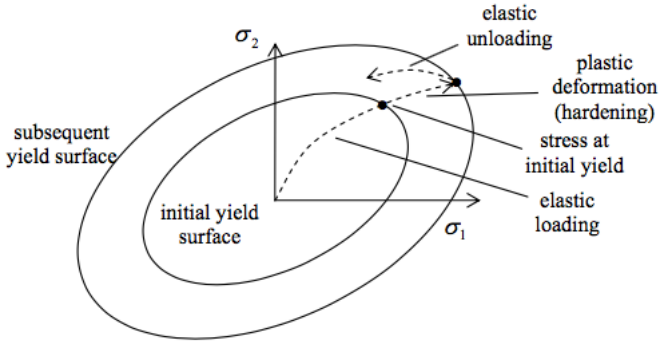

Figure 3.4. Isotropic hardening.

According to the isotropic hardening, if the yield strength is initially equal during tension and compression (yield surface being symmetric about stress axis), they continue to remain equal as

the yield surface develops with plasticity. Kinematic hardening, on the other hand, is used for cases where hardening in tension will lead to a softening in a successive compression. In this type of hardening, the yield surface maintains its shape and size but translates in space, as shown in Figure 3.5.

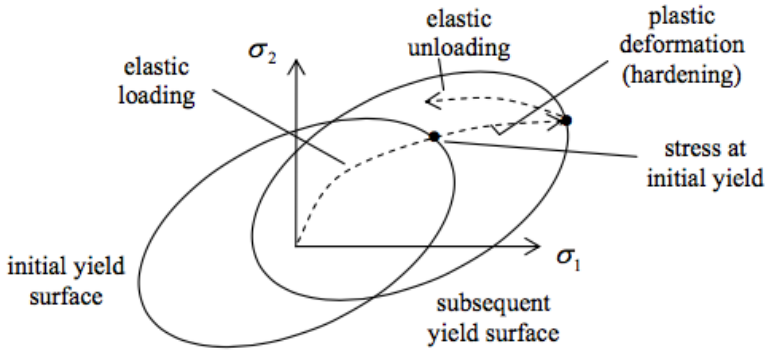

Figure 3.5. Kinematic hardening.

The yield is taken into account in the form of a function that is shown in equation 11.

$$f(\sigma_{ij}, K_i) = f_0(\sigma_{ij} - \alpha_{ij}) = 0 \quad \text{.....} \quad (11)$$

In this case, stress α_{ij} is taken as the hardening parameter. This is also known as shift stress, which means that the yield surface shifts by α_{ij} relative to the stress-space axes.

3.6 Implicit and Explicit Analysis

Incremental steps for loading and displacements are needed in geometrical and material nonlinear analysis. As explained in Chapter 3.4, the end of each increment step changes the structural geometry as well as causes the material to yield. Each of these geometric and material nonlinearities need to be taken into account while updating the stiffness matrix for the analysis' next increment step.

For an Explicit FEM analysis, the updating of the stiffness matrix is done at the end of each increment. Following this, the system is subjected to the next increment of load or displacement. Smaller increments in the analysis typically lead to more accurate solutions, which could lead to time-intensive computations. If the number of small increments are inadequate, the solution tends to be incorrect. This may lead to a problematic situation where computational resources have to

be balanced with the level of accuracy of results. The main drawbacks of Explicit analysis are that it is unable to solve problems of cyclic loading and cannot enforce equilibrium between the internal forces of structure and the external loads.

In order to overcome the drawbacks of Explicit analysis, Implicit FEM analysis also carries out Newton-Raphson iterations after each increment that enforces equilibrium of the internal forces of structure with the external loads. This leads to results with more accuracy and the ability to work with bigger increment steps. In addition, Implicit analysis can handle problems of cyclic loading. A significant drawback of this type of analysis is that it can be computationally expensive as a result of using Newton-Raphson iterations for updating and reconstructing the stiffness matrix for each iteration. To avoid this, Modified Newton-Raphson methods can be used.

In the case of static loading (quasi-static loading), it is preferred to have big time steps so that the solution can be obtained in less number of computationally-intensive steps. Hence, implicit analysis is generally used for such situations. In the case of dynamic loading such as impact tests, however, explicit analysis is preferred. Selection between Implicit and Explicit analysis is mainly a trade-off between a more accurate and a computationally lighter (faster) analysis.

4 Scope of work

Ship structures at multiple hull locations have been modelled with shell elements in the Finite Element solver Abaqus. This has been followed by modeling the ice loads with different scenarios. The main objective has been to determine the critical loading scenarios of framing and plating when subjected to different load distributions compliant with measured load shapes. Applied loads will be higher than defined in the rules in order to determine safety margins against three different capacity levels for the serviceability limit state (SLS): 1) yield (current design criteria in FSICR); 2) 3-hinge mechanism and 3) permanent deformation according to DNV.

In this thesis, two different ships have been considered (MT Uikku and SA Agulhass II), with three hull locations (bow, mid and stern) and for three ice classes (IA Super, IA, and IB). In total, 15 different models have been constructed. Matlab scripts have been used to calculate the scantling dimensions used for creating these computational models according to FSICR. The analyses are performed with ships for which long term full-scale ice load measurements are available (Kurmiste, 2016 and Kaldasaun, 2010). Therefore, general conclusions can be drawn regarding the response of the structures. This data also includes ice conditions, which is further categorized based on ice thickness and ice concentration. Although outside of the scope of present thesis, this allows one to associate each load level with particular ice conditions and its probability of occurrence. Practical significance is that one can assess the validity of the preliminary assessment of the operational restrictions for ships ice-strengthened in accordance with different Finnish-Swedish Ice Classes. Furthermore, the calculated safety margins with respect to different capacity levels will help in tackling the over-dimensioning of ships by formulating lighter scantling requirements while maintaining the safety level with respect to the ice loads.

This thesis includes work on the following areas:

1) FSICR for ice-strengthening of ships navigating in the Baltic sea have been reviewed, with emphasis on requirements to plating & framing. In addition, the state of the art in ice-load definitions have been studied.

2) Ship structures designed in accordance with ice class rules have been analysed in the Finite Element solver Abaqus, and operational restrictions of Uikku and Agulhass ships have been validated.

3) The sensitivity of different models to load application have been studied. For this, ice-loading scenarios have been varied for the different models and its effect on the structural response have been investigated.

4) On the basis of these Finite Element models, three capacity levels have been investigated. Safety margins against three different capacities have been calculated by comparing the numerical simulation results with FSICR design load.

5 Methods: Approach for modeling ice loads

5.1 Non-uniform pressure patch

Two different approaches are used to represent ice loads. First, a non-uniform pressure patch (NUPP) is used in the simulations that leads to larger deformations with the smaller line load (Kõrgesaar et al. 2017), see Figure 5.1. This makes the approach less conservative compared with present design approach of FSICR. Pressure across single frame spacing is uniform and constant, but decreases exponentially outside of this area. The formulation for exponential reduction is obtained from the *line load – load length* relationship given in Figure 2.3. Specifically, the SA Agulhass II stern measurement are used. The load shape resulting from this curve has been calculated for both MT Uikku (Figure 5.1) and SA Agulhass II ships, and will be used throughout this thesis.

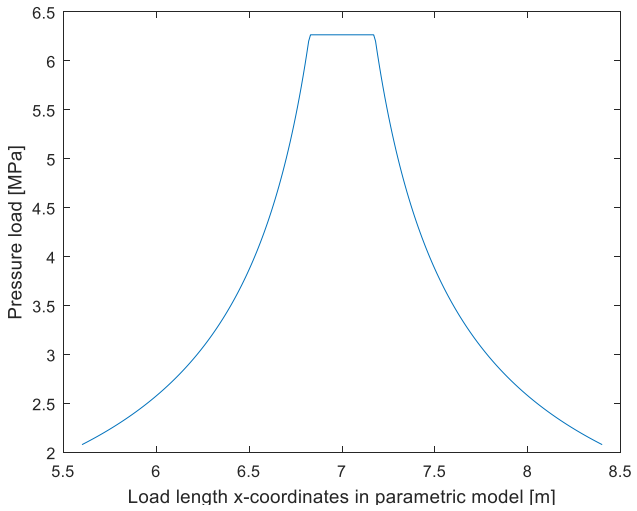

Figure 5.1. Loading scenario to be applied in Uikku bow (IA Super) parametric model

In the finite element models, the non-uniform pressure is applied on a patch spanning from webframe to webframe. This load patch is vertically located at the center of the stringer spacing, see Figure 5.2.

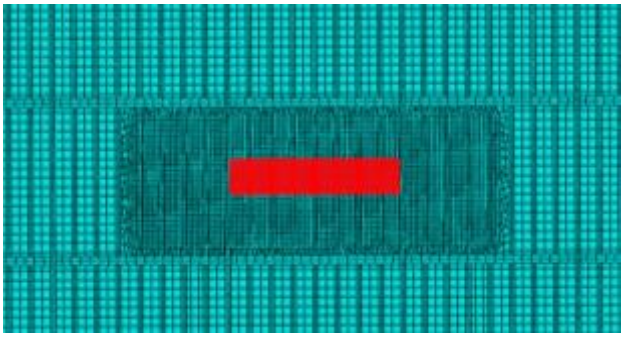

Figure 5.2. A representation of the load patch location (from webframe to webframe) used in FE analysis of models

The approach of using NUPP is believed to have following advantages compared to FSICR design approach:

1) It considers the local character of the load, similar to FSICR.

2) It considers the fact that load is likely distributed over a longer length than the frame spacing.

Hence, the NUPP used in this thesis provides a middle ground between FSCIR approach (pressure applied over one frame spacing) and an approach of using constant pressure over the width of the webframe.

To create this load shape (Figure 5.1) from the curve of Agulhass stern, the variables in equation 2 need to be substituted for each ship. For Agulhass stern, s (frame spacing) is 0.4 m, while C and a (unknown parameters) are obtained as 1989 kN/m and 0.733 respectively. After defining these values in equation 2, the following function (l_c is load length) has been obtained:

$$q = 1989\left(\frac{lc}{0.4}\right)^{-0.733},\ q = 1.0161(lc)^{-0.733}.10^3\ kN/m \ldots\ldots\ldots\ldots\ldots\ldots \quad (12)$$

The equations derived in this section have to be ultimately defined in the finite element solver Abaqus/CAE, which requires loading to be defined as pressure load. For this, line load q (represented by equation 12) has been divided by load height h to convert it into pressure load. For each ice class from IA Super to IB, load height h has been taken as 0.35 so that the pressure load is applied on an equal area across ships of different ice classes. Henceforth, the follow equation is obtained for a ship of ice class IA Super, IA or IB (pressure across ice classes remain same):

$$\text{Pressure} = \frac{q}{h} = \frac{q}{0.35} = 2.903\ (lc)^{-0.733}.10^3\ kN/m^2 \ldots\ldots\ldots\ldots\ldots\ldots \quad (13)$$

The above cases (Case Study I in Chapter 7) enable to assess the changes in structural carrying capacity between ships of different ice classes, which can be seen in chapter 7. As the height of loading is the same for all ice classes, the decreasing load as a function of length will remain the same and the magnitude of the loading is varied. Additional cases (Case Study II in Chapter 7) have also been defined in which the height of the pressure patch was changed according to ice class. These cases are aimed at testing the effect of load height h on the structural response of the ship hull. Consequently, the magnitude of equation 13 will change when load height h is varied (the differences in magnitude of pressure have been summarized in Table 5.1).

Table 5.1. Difference in the magnitude of Decreasing load function when load height is varied in FEM models

Parametric model	Decreasing load function [MPa]
Uikku bow, midship and stern IAS (h = 0.35)	2.903 $(lc)^{-0.733}$
Uikku bow, midship and stern IA (h = 0.30)	3.387 $(lc)^{-0.733}$
Uikku bow, midship and stern IB (h = 0.25)	4.064 $(lc)^{-0.733}$
Agulhass bow, midship and stern IAS (h = 0.35)	2.903 $(lc)^{-0.733}$
Agulhass bow, midship and stern IA (h = 0.30)	3.387 $(lc)^{-0.733}$
Agulhass bow, midship and stern IB (h = 0.25)	4.064 $(lc)^{-0.733}$
Agulhass bow, midship and stern IB (h = 0.25)	7.956 $(lc)^{-0.733}$

As the pressure load is inversely proportional to load height, it can be seen in the above table that the magnitude of pressure loads – both uniform and decreasing function - increases with decrease of load height.

Non-uniform pressure patch corresponding to Uikku bow IA Super design is plotted in Figure 5.1. To replicate this load shape in parametric finite element models, the equation 13 is defined in FE software Abaqus/CAE using User Defined Subroutine. This is discussed in more detail in the next section.

5.2 Uniform pressure across the web-frame

This case will be implemented as shown in Figure 5.3. Uniform FSICR design load will be applied on the load patch locations along the webframe spacing (Case Study III in Chapter 7). This case study will act as a comparison point for the NUPP and will be done for multiple ice classes.

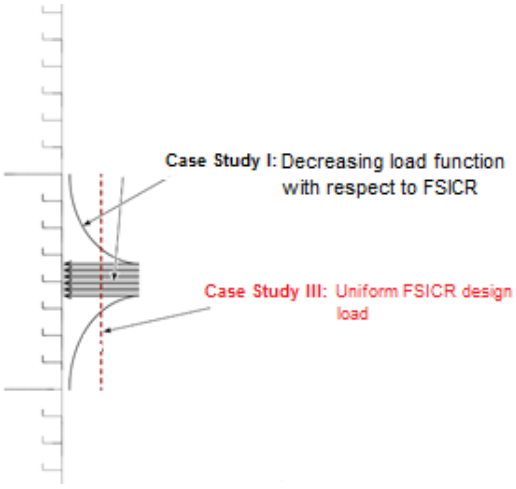

Figure 5.3. Pictorial description of the case studies (loading scenarios) to be investigated in this thesis

6 Case Studies

As a case study structure, bow, midship and stern regions of ships MT Uikku and SA Agulhass II were selected. The ship scantlings were constructed according to FSICR ice classes IA Super, IA and IB. Details of each ship are documented in this chapter. For finite element modelling in this thesis, the FE software Abaqus/CAE 6.14-1 has been used. The details of the finite element simulations conducted in this project have also been detailed in this chapter. In this thesis, there are four major case studies, with Case Study I having two further case studies. Each of these are discussed in detail in Chapter 7 along with their results.

6.1 Structural dimensions for ships used in case studies

For the case studies conducted in this thesis, two ships – S.A. Agulhass II and Uikku MT – have been selected. S.A. Agulhas II was built to be classified as Polar ice class PC5 and the hull was constructed in accordance with DNV ICE-10. For the structural scantlings of bow, mid-ship and stern, T-frames are used. The frame span for the bow is 2.065 and 1.1 m for the mid-ship and stern. The frame spacing and webframe spacing for bow, mid-ship and stern is 0.4 and 2.4 m respectively. Three areas of the starboard side of the hull were instrumented with strain gauges when it was under construction in 2011/2012 (as shown in Figure 6.1). Ice-induced loads were determined by instrumenting the upper and lower parts of the frame with V-shaped strain gauges, which measured the shear strains occurring in the frame. The instrumentation was applied to two adjacent frames at the bow, three adjacent frames at the bow-shoulder and four adjacent frames at the stern-shoulder.

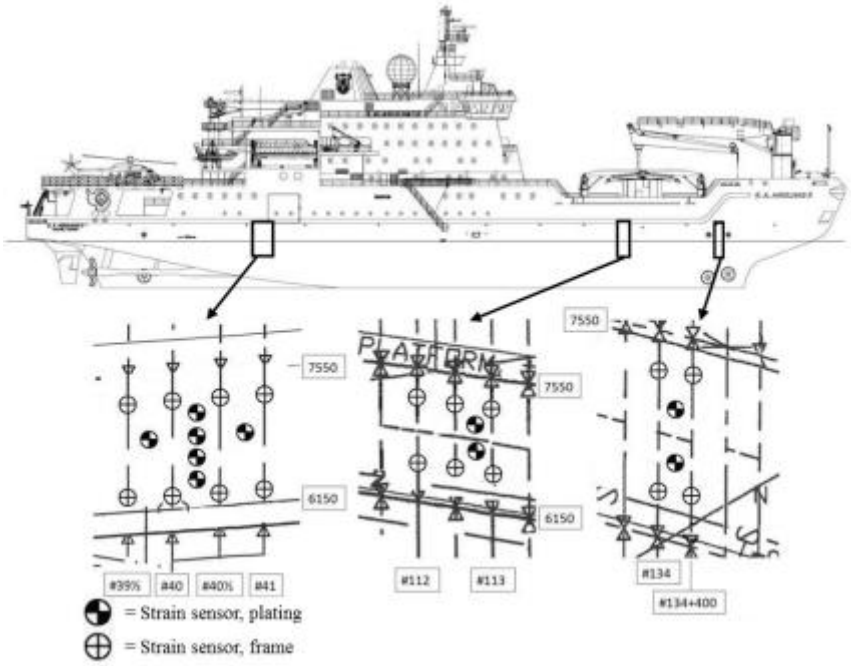

Figure 6.1. Instrumented areas of S.A. Agulhass II

M/T Uikku, on the other hand, is classified by DNV as class +1 A Tanker for Oil and according to FSICR, it is classified as ice class IA Super. The frame span for the bow is 2 m, 2.92 m for the bow-shoulder and 1.22 m for the stern, the frame spacing is 0.35 m. To measure ice-induced loads on the hull, the frames of the ship were fitted with instrumentation in the bow, bow-shoulder, mid-ship and aft area (as shown in Figure 6.2). The ice loads were evaluated by measuring shear strains at roughly the neutral axis of the frame. The main particulars of both the ships are summarized in the table below.

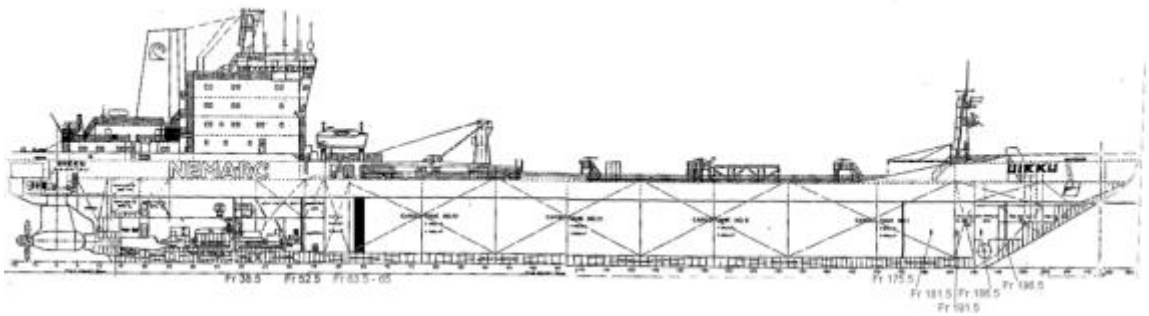

Figure 6.2. Instrumented areas of M.T. Uikku

Table 6.1. Main particulars of S.A. Agulhass II and Uikku MT

S.A. Agulhass II		Uikku MT	
Length, bpp.	121.8 m	Length, bpp.	150 m
Breath, mould.	21.7 m	Breath, mould.	22.2 m
Draught, design.	7.65 m	Draught, design.	9.5 m
Deadweight at design displacement	5000 t	Deadweight at design displacement	15748 t
Displacement	13632 t	Displacement	22654 t
Speed, service	14 kn	Speed, service	17 kn
Propulsion power	9 MW	Propulsion power	11.4 MW

The input dimensions for both ships are summarized in the table below.

Table 6.2. Input dimensions of the ships for case studies

	Location	Frame spacing	Webframe spacing	Stringer spacing
S.A. Agulhass II	Bow	0.4	2.4	2.065
	Mid-ship	0.4	2.4	1.1
	Stern	0.4	2.4	1.1
MT Uikku	Bow	0.35	2.8	2.5
	Mid-ship	0.343	2.1	2.1
	Stern	0.35	2.1	2.5

6.2 Finite element simulations

For finite element modelling in each of the case studies, the FE software Abaqus/CAE 6.14-1 has been used, which is capable of handling large-scale deformations and failure scenarios. In the geometrically nonlinear analyses, loading is increased by small steps and carried out by Newton's method, as shown in chapter 3.4. All the nonlinear FEM simulations have been carried out using CSC's Taito server (capacity of which consists of eight CPUs with 32 GB RAM each).

6.2.1 Finite element model

The scantlings of Agulhass II and MT Uikku have been calculated by defining the FSICR equations and each ship's input dimensions (Kujala, 1991) in Matlab scripts (see Appendix 1). During scantlings calculations, the brackets have been assumed to be non-existent in the models. Using the obtained scantlings from Matlab, Abaqus/CAE python module has been used to build parametric finite element models of both ships. These parametric features provide the swift ability to examine the effects of design changes. Using parametric modeling, a finite element model and its mesh characteristics can be completely defined in terms of parameters or variables. A finite element parametric modeling method of ships overcomes the time-consuming aspect of finite element analysis pre-processing. Thin shell elements (S4R) have been used to model frames, plating, web frames, flanges and stiffeners. The assembled model of the ship structure for Uikku bow is shown in Figure 6.3. The large grillage structure has five frame spans/webframe spacings in both transverse and longitudinal directions.

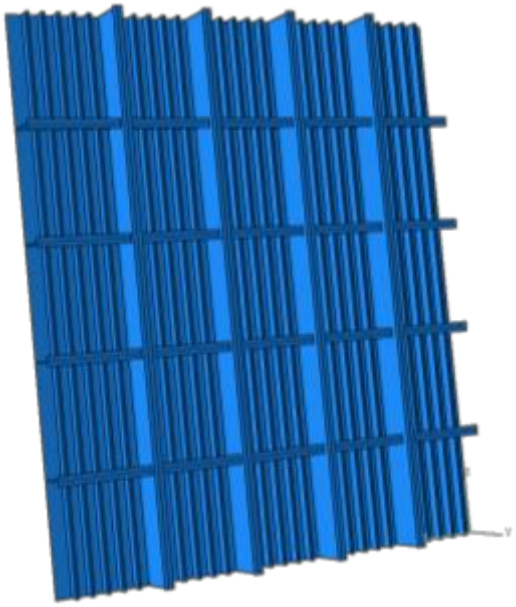

Figure 6.3. Structural geometry and boundary conditions

The parametric modeling works with the help of three python scripts. The script file *variables_script.py* defines the main model variables. It is here that the main parameters can be varied and the relevant changes can be seen in the parametric model. The most important parameters are shown in the Table 6.3 below (Python scripts can be found in Appendix 3). The second script file *TRAFI_Grillage_script.py* builds the model itself. In addition to these two files,

the *Main_script.py* file is used to call the model creating scripts. As a result of running these scripts in Abaqus/CAE, the parametric model and its associated input files are created.

Table 6.3. Main variables of the parametric script

	Description of the parametric variable
Plates and frames	
frame_flange	Width of the frame's T-flange (defined as half of the total width)
tframe	Thickness of frame
tframe2	Thickness of T-flanges of frame
fh	Frame height
tp	Thickness of plate
st_w	Webframe spacing
Stringers	
girderH	Stringer height
girder_flange	Width of the stringer's T-flange (defined as half of the total width)
tgirder	Stringer thickness
Tgirder2	Flange thickness
Webframes	
webheight	Webframe height
webframe_flange	Width of the webframe's T-flange (defined as half of the total width)
tweb	Webframe thickness
tweb2	Flange thickness
depth	Stringer spacing dimensions

For the simulations performed on the parametric models, Implicit FEM analysis has been used. This has been done primarily because of the limitations in this thesis (non-contact problem and no fracture/crack growth), for which the Implicit analysis is efficient. Implicit analysis enforces equilibrium of internal structural forces with the applied load after each incremental load step. In addition, Implicit analysis tends to be more accurate than Explicit analysis in which the solution loses accuracy if the number of increments are insufficient. However, Implicit is also computationally more expensive than the Explicit method.

After the input files have been obtained, non-linear analyses is run in Abaqus/CAE. From the analyses, the force data is obtained by summing the Out of plane (RF2) Force components in the boundary nodes. The displacement data is extracted over a patch and the node that experiences the largest deformations is selected. The entire process followed in this thesis for running analyses has been summarized in a flow chart (Figure 6.4).

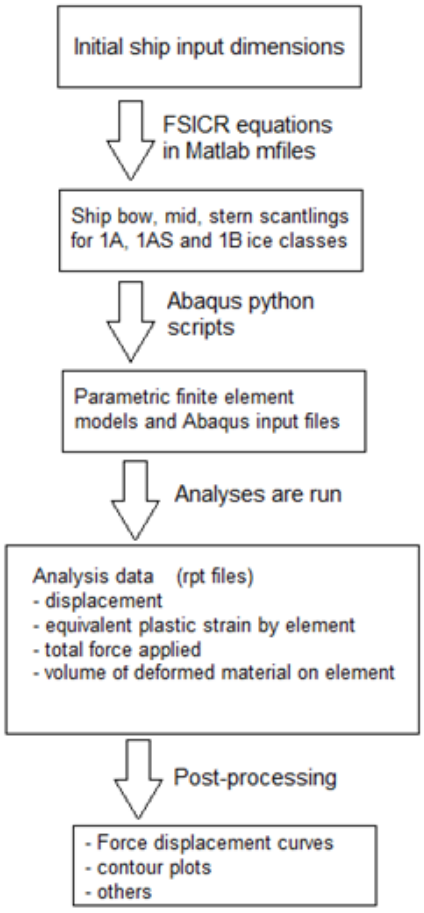

Figure 6.4. Flow chart showing the process followed for analysis of ship parametric models in this thesis.

The current FE analysis requires two nonlinear assumptions:

1. Material nonlinear: the plasticity of the steel is considered.
2. Structural nonlinear: under extreme ice loading, the structures may exhibit large deformation.

6.2.2 Material modelling and properties

Experimentally obtained engineering stress-strain relation of the material has been used for material modelling in this thesis. This yield stress-plastic strain curve for material behaviour was measured as part of the other ongoing project at the Marine Technology research group. The material that has been used is a standard structural steel S235JR with following minimum values for mechanical properties defined in standard EN 10025-2: ReH = 235MPa, Rm = 360…510MPa and elongation after fracture of 26%. The true stress-strain curve shown in Figure 6.5 was

determined with 3mm thick tensile (dog-bone) specimens. 285 MPa yield strength steel has been used in FE models of all the ship cases. The other material properties have been summarized in the table below.

Table 6.4. Material properties used for computational analysis

Material	Young's modulus	Poisson's ratio	Fracture strain	Mass density
Steel	200 GPa	0.3	0.35	7850

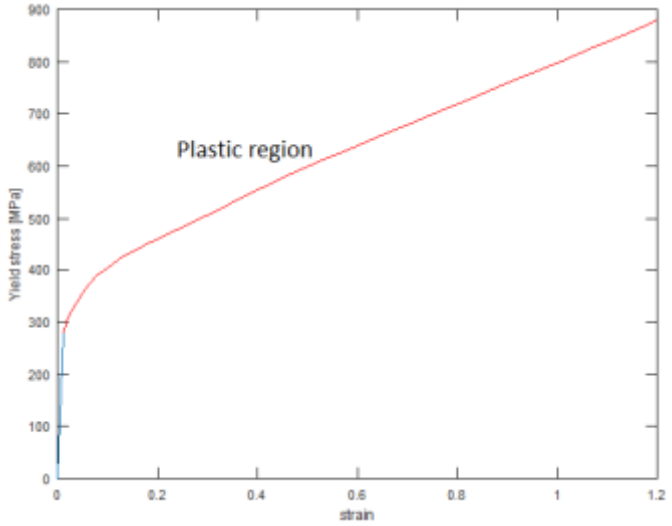

Figure 6.5. Yield stress-plastic strain curve for the material used for FE model.

6.2.3 Boundary conditions

At each span, a stringer or webframe is provided which gives the necessary support at that location (Figure 6.3). Although smaller stiffening members such as brackets are not modeled, the grillage structure is assumed to be a good representation of the stiffening in real ship side structure and suitable for comparative analysis. The transverse and longitudinal ends of the large grillage structure have translations fixed and rotations free. Since, the location of fixed points are far from the point of application of the loading (on the center frame/webframe), the boundary conditions have effect on the response where load is applied. Such a large grillage is considered to be an accurate representation of a real ship structure.

6.2.4 Loading and load locations

The load shape to be applied on the refined load patch has been explained in Chapter 5.1. In the analyses, the load patch location has been positioned between two webframes. The load patch location is approximately demonstrated in Figure 5.2. In some of the analysis cases, larger

deformations were leading to an error aborting the analysis before initiation. To solve this problem, the refined mesh area needs to be slightly increased in the height direction.

The analysis step definition in the models have been done such that the step time is 2 seconds and the maximum number of increments is 500, with non-linear geometry parameter (nlgeom) switched on. Static step is used in the simulations and the pressure application has been be done through a V-shaped amplitude curve (as shown in Figure 6.6a) that reaches the maxima in the middle of analysis and then comes back to zero. This process simulates loading - unloading and hence, the permanent deformations on the structure can be determined. Further, the loading scenarios and magnitude have been defined in Abaqus/CAE using a user-defined subroutine DLOAD and VDLOAD. The VDLOAD subroutine has been used only in the case for Uikku mid IA as the Implicit analysis was resulted in an error for this specific case. With the use of explicit VDLOAD subroutine, the response is slightly stiffer initially than what was obtained with implicit. Preliminary simulations showed that magnitude of the pressure according to NUPP defined with equation 13 and shown in Figure 5.1 is too small to inflict significant damage to structure. Since the objective was to investigate the capacity of the structure, magnitude of the NUPP was increased. Figure 6.6b shows the pressure profile of non-uniform patch after multiplying the initial NUPP with factors 2 and 3. Eventually, a multiplier of 3 has been used for load definition in each of the analysis cases of this thesis.

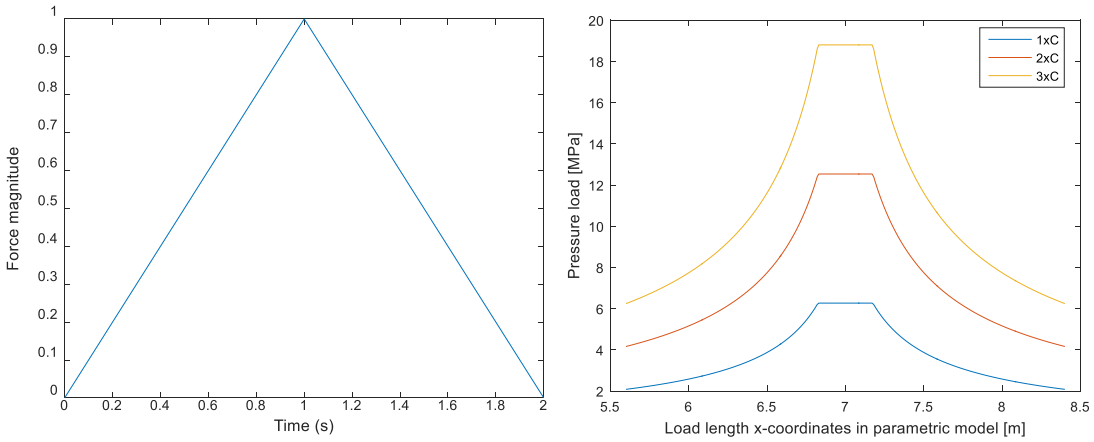

Figure 6.6 (a). Loading on the model following a V-shaped amplitude curve. (b). Plot showing effect of multiplier on the load shape applied in FE models

6.2.5 Element type and Mesh size

In the models created for this thesis, four-node quadrilateral shell elements have been used. These elements are reduced integration S4R shell elements with 5 integration points through thickness and stiffness based hourglass control. They possess six degrees of freedom on each node.

The FE models are meshed with a size of 0.15, with an additional refinement of 0.05 performed in the load patch region. This refinement has been conducted as it is useful to place coarse meshes at the less important areas of a model. The 50 mm mesh was deemed sufficient by preliminary comparative analysis with 25 mm mesh, which did not show any appreciable changes in response at the deformation levels that are of interest in this analysis. By doing so, the analysis time may be significantly decreased. On the other hand, a sufficiently refined mesh in the pressure application locations is necessary for accurate computational results in Abaqus/CAE. If a coarse mesh is produced in those critical regions, it can result in lower accuracy in analysis especially at large deformations, folding and buckling. Hence, the patch region is modeled with a fine mesh since those areas are both for application of loads and are of interest to observe the FE results.

6.2.6 Analyzed locations along ship hull

In order to determine how capacity depends on the location along the ship hull three different regions have been considered: bow, mid and stern. Furthermore, three ice classes are considered IAS, IA and IB. In total, there are 15 case studies that have been performed as shown in the flowchart in Figure 6.10.

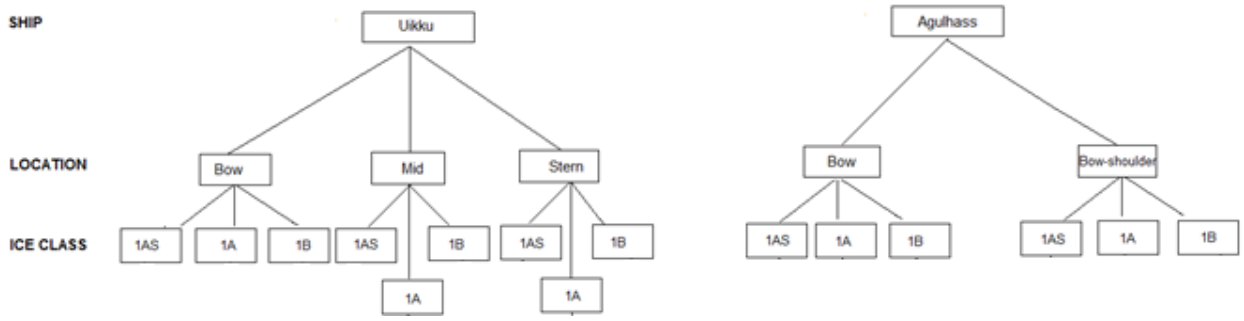

Figure 6.10. Flow chart representing the different cases that have been analyzed

7 Results and Discussion

In this Chapter, four different case studies have been discussed which focus on the response of finite element models when subjected to different ice loading scenarios. This has been followed by calculating three capacity levels for all 15 finite element models of each case study and drawing comparisons among the different cases. Further, safety factors for each of these cases have been calculated, which are among the final results.

7.1 Case study I: Response to Non-uniform pressure patch (NUPP)

The discussion in this section is focused on the response of ship structures when loaded with the NUPP described in Chapter 5.1. The Finite element analyses in Abaqus/CAE for all the cases resulted in displacement data, force data, equivalent plastic strain data and volume of deformed material. After plotting the force vs. displacement data for models, the Figures 7.1 and 7.2 were obtained.

In this case study, load height corresponds to ice class: the IAS ice class models have been loaded with a load height of 0.35 m, whereas the IA and IB class models have been loaded with a load height of 0.30 m and 0.25 m respectively. Since the pressure is derived from the same line load function (Chapter 5.1), the total load applied in simulations is exactly the same, but pressure for lower ice classes is higher.

The NUPP has been applied on the width of webframe. Conventionally though, long term measurement data has usually been available for single frame. Hence to fully utilize the present results at hand and to make comparisons with measured data, load on single frame is more relevant than total load. Hence, load-displacement curves show the load on a single frame.

Figures 7.1 and 7.2 show that lower ice-class ships exhibit larger deformations that is consistent with the present design approach. This is evident from the shift of the curve towards right for lower ice class. In case of Agulhass this reduction in capacity is very similar along different regions of ship hull: bow, midship or stern. The same applies for Uikku with exception of midship region. There, the capacity of IA Super and IA design is very similar.

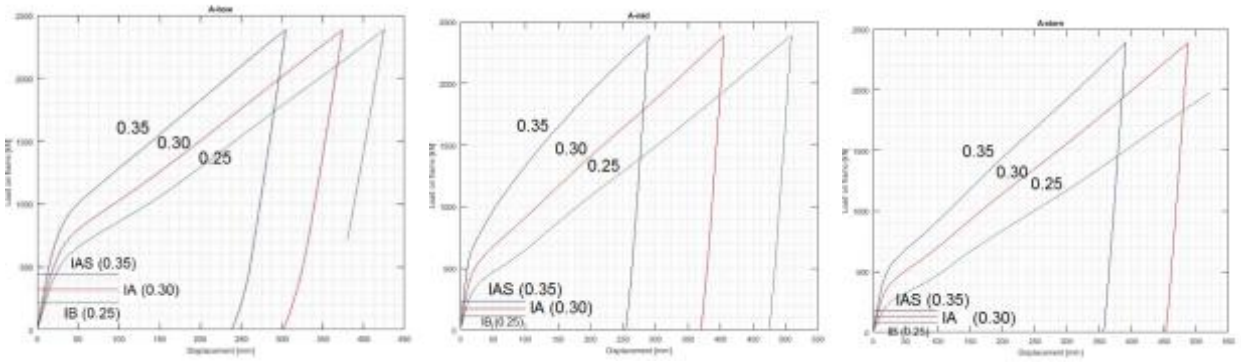

Figure 7.1. Load-displacement for Agulhass bow (left), mid-ship (middle) and stern (right) for three ice classes: IAS (load height 0.35), IA (0.30) and IB (0.25). The FSICR design load is plotted for comparison.

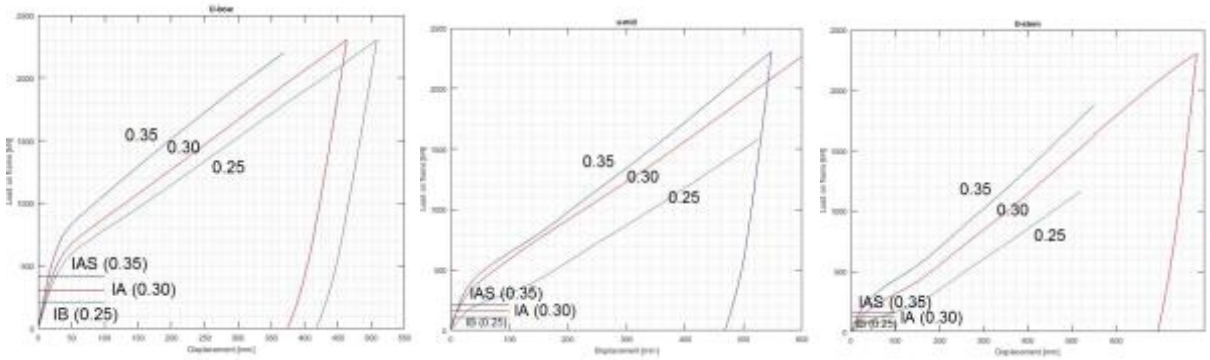

Figure 7.2. Load-displacement for Uikku bow (left), mid-ship (middle) and stern (right) for three ice classes: IAS (load height 0.35), IA (0.30) and IB (0.25). The FSICR design load is plotted for comparison.

Figures 7.1 and 7.2 are focused on the effect of ice class on capacity. To understand how hull region affects capacity, load-displacement plots for different regions along hull are given for one ice class in Figure 7.3. These plots represent three locations – bow, midship and stern – for ships of the same ice class IA Super. Again the decrease in capacity is consistent with the design approach whereby bow should be considerably stronger than stern.

For the case of Agulhass ship, however, bow and midship locations have similar deformations when subjected to the same ice load. One possible reason for this behavior is because the webframe spacing on Agulhass bow and mid is equivalent - with a magnitude of 2.4m. On the other hand, for the case of Uikku ship, the webframe spacing of bow and midship are not equivalent, which also leads to dissimilar deformations when subjected to the same ice load. Nonetheless, in the corresponding stern location for Agulhass ship, increased displacements can be observed when the same ice load is applied.

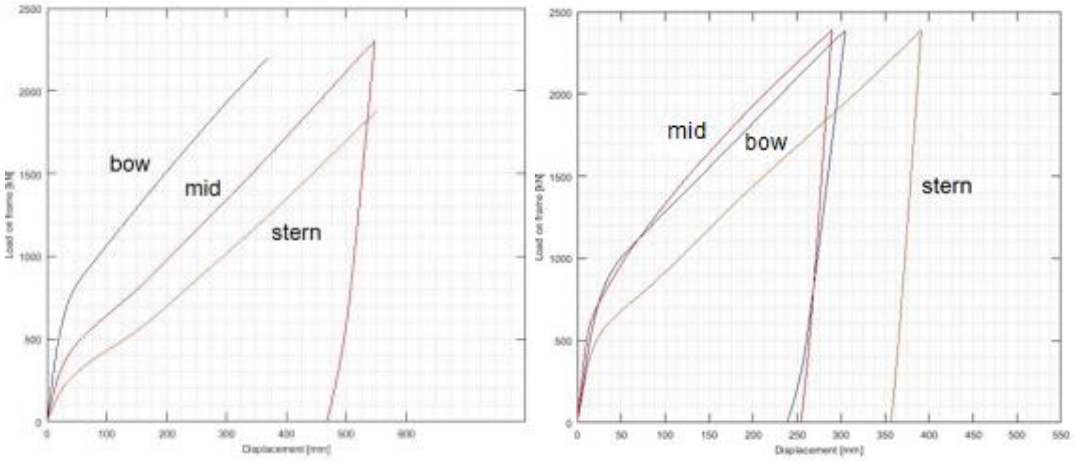

Figure 7.3. Load on frame (kN)-displacement (mm) plots of Uikku and Agulhass for different locations Bow (blue), Mid-ship (red) and Stern (yellow) for IAS ice class and load height 0.35.

A contour plot of Mises stress for Uikku bow IAS has been shown in Figure 7.4, which demonstrates the effect of the load shape (Figure 5.1) on the plating and frames. NUPP imposed on the structure causes highest stresses in the middle frame while it decreases outside of this area.

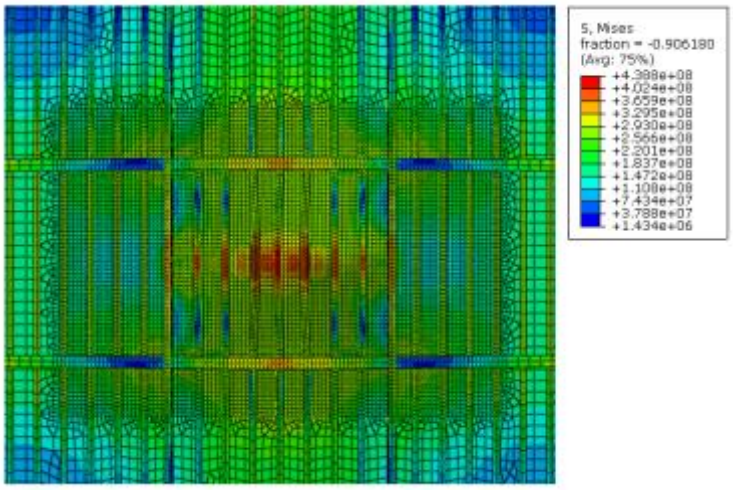

Figure 7.4. Contour plot of Mises stress for Uikku bow IA Super model

In order to get deeper insights into the plasticity behavior of these structures, their first fiber yield have been analyzed. This has been demonstrated in Figure 7.5 for Uikku ship IA Super and subjected to NUPP. Three points have been selected on the force-displacement curve. First point represents the force exerted before reaching the yield limit, second point represents the force applied at the yield point and the third point denotes the force after the structure has gone plastic.

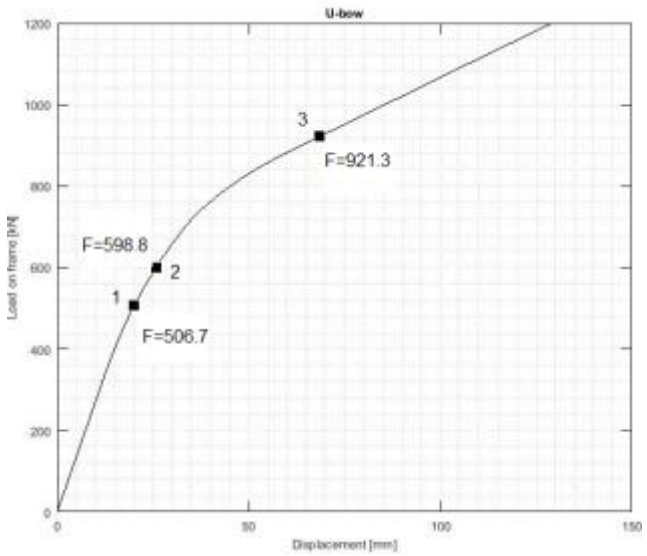

Figure 7.5. Load on frame (kN)-displacement (mm) plots curve for bow location of a Uikku ship built with ice class IAS and subjected to decreasing load function with load height 0.35 (Force 'F' units in kN)

In the Figure 7.5, three points have been marked in which the second point (Force = ~600 kN) is the point of yield. For each of the above points, the Mises stress and in-plane principal plastic strain have been visualized at those time instances using the contour plots shown in Figure 7.7 and 7.6 respectively. The contour plots in these two figures are obtained for the center frame located in the middle of the webframe.

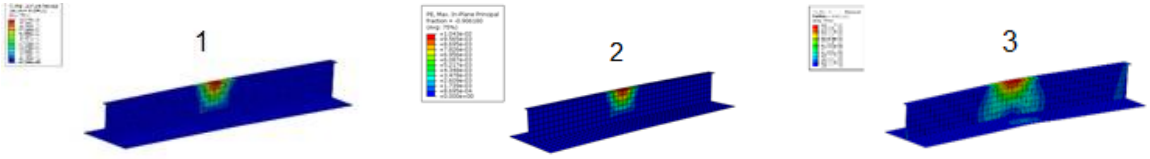

Figure 7.6. Contour plots of In-plane principal plastic strain for Uikku's bow location (IAS ice class and load height 0.35). Left plot is for point 1 (F=506.7). Middle plot depicts the plastic strain for point 2 (F=598.8), which is the yield limit. Right point is for point 3 (F=921.3). The frame that is discussed here is the one at the center of the webframe spacing.

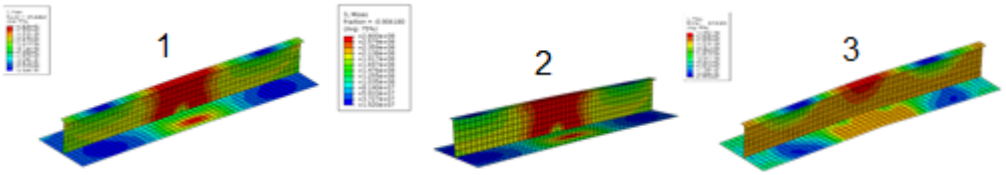

Figure 7.7. Contour plots of Mises stress for Uikku bow's location (IAS ice class and load height 0.35). Left plot is for point 1 ((F=506.7). Middle plot depicts the plastic strain for point 2 (F=598.8), which is the yield limit. Right point is for point 3 (F=921.3). The frame that is discussed here is the one at the center of the webframe spacing.

7.1.1 State of yield based on Finite Element Simulations

To draw some conclusions based on the previous contour plots in Figures 7.6 and 7.7, two cases have been designed and their results have been examined.

7.1.1.1. Case 1 (comparison between plasticity behavior of different hull locations): Uikku bow, mid and stern IA Super

When the force-displacement plots for different hull locations are compared, bow is found to exhibit plastic deformations at higher loads than the mid and stern counterparts, as shown by points 1, 2 and 3 in Figure 7.8. This behavior is in-line with that observed earlier in Figure 7.3 which showed that mid-ship and stern locations undergo much higher deformations than the corresponding bow location when subjected to the same loading scenario. Furthermore, comparison of yield to design load shows that design loads are ~70% of the yield load for the bow location. To recall, design point of FSICR is reaching yield at least once per winter. The safety margin against yield for the mid and stern locations are ~62% and ~65% respectively.

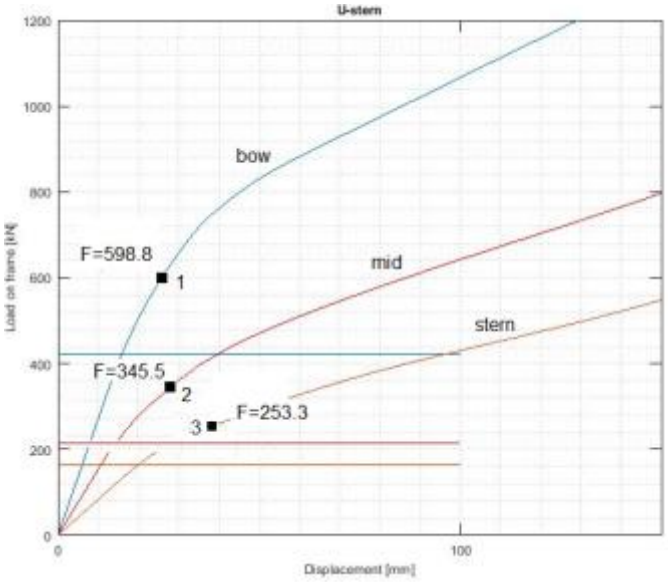

Figure 7.8. Case 1: Load - displacement curves for Uikku ship's bow (blue), mid (red) and stern (orange) locations. Ice class is IA Super. Points 1, 2 and 3 represent the yield limit for each curve. The FSICR design loads have also been plotted for each location (blue, red and orange).

When the contour plot results for plastic strain and Mises stress are compared at points 1, 2 and 3, there is no substantial difference between strains and stresses (Figure 7.9 and 7.10). This validates that plasticity has been reached at those points for each of the respective structures.

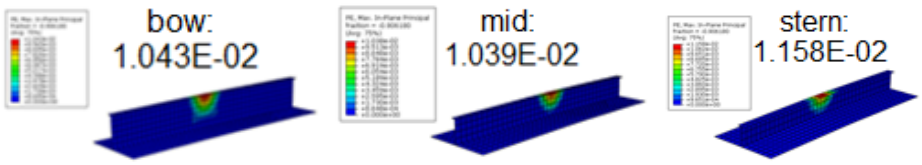

Figure 7.9. Contour plots of In-plane principal plastic strain for Uikku bow (left), mid (center) and stern (right) locations. The frame that is discussed here is the one at the center of the webframe spacing.

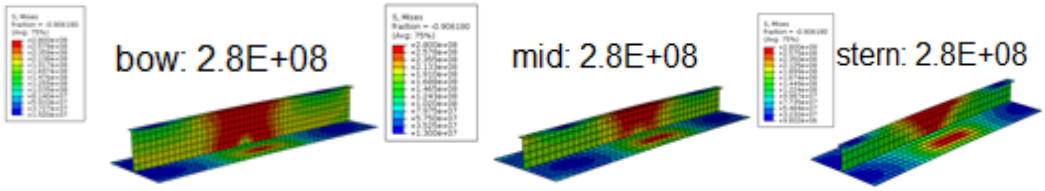

Figure 7.10. Contour plots of Mises stress for Uikku bow (left), mid (center) and stern (right) locations. The frame that is discussed here is the one at the center of the webframe spacing.

7.1.1.2. Case 2 (comparison between plasticity behavior of different ice classes): Uikku bow IA Super, IA and IB

For case 2, different ice classes for the same location are compared in Figure 7.11. The IA Super case is shown to yield at higher load when compared to IA and IB. The obvious reason for this behavior is that the higher ice classes (IAS and IA) are heavily dimensioned as compared to IB. Because of this, increased ice loading is required to exhibit plastic behavior in those cases. Inception of plasticity might be also affected by the more concentrated pressure applied on lower ice class structures to manifest the same line load in each ice class. The effect of load height be further investigated in the next section.

Similar to Case 1, the contour plots for Case 2 show that there is no noticeable effect on plastic strain and Mises stress with change in ice classes of Uikku bow.

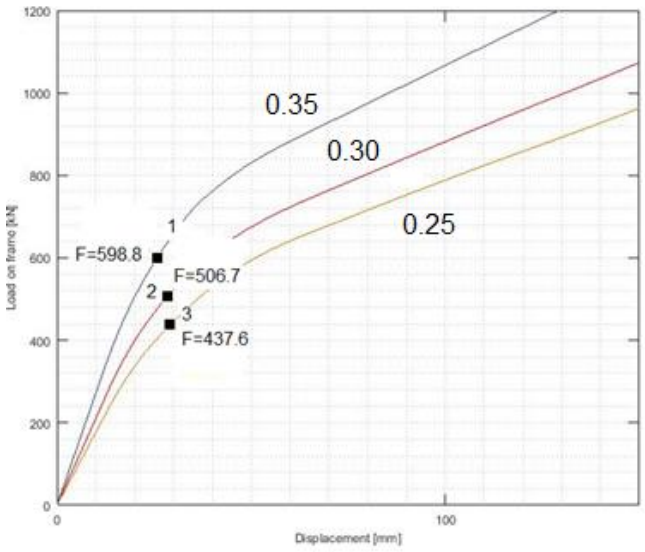

Figure 7.11. Case 2: Load - displacement curves for Uikku's bow location for ice classes IA Super (blue), IA (red) and IB (orange). Points 1, 2 and 3 represent the yield limit for each curve.

7.1.2 Comparison of the capacity for Case study I

One of the objectives of this thesis was to investigate different levels of capacity, and to determine safety margins for those. These three capacities are: 1) yield load; 2) three hinge pressure load; and 3) load for permanent deformation according to serviceability limit state (SLS) defined by DNV, frame spacing (s/12). This comparison is summarized in Tables 7.1 and 7.2 for MT Uikku and SA Agulhass II ships respectively. The permanent deformation in the structures were determined by assuming that slope of the unloading curve remains the same due to the permanent nature of plastic strains. Thereby, the unloading portion of the curve is shifted towards left along the x-axes so that the permanent set would be exactly equal to the reference permanent displacement. However, as some simulations prematurely ended as convergence was not reached the unloading cycle was not available, see Figure 7.1 and Figure 7.2. In those cases, the unloading cycle from different simulations was used (e.g. for Uikku bow IA Super unloading curve of Uikku bow IA was used). This will have effect on the load value corresponding to permanent deformation as the slopes are different for different cases. This issue will be highlighted further when results are discussed.

From the three capacity levels, yield is reached first. The last line in Table 7.1 gives the ratio between design load to yield load. The closer this ratio is to 100% the more successfully the structure follows the design intention which is reaching the state of yielding when subjected to

design pressure. In this respect, all ice classes seem to be over-dimensioned. The IA Super class seems to be most effectively designed with average ratio of 65%. In contrast, for the IB class design load is only 50% of the yield load.

As expected, structures have more reserve capacity against permanent deformation limit. The ratio of design load to permanent deformation load follows the same trend as the ratio corresponding to yield – IA Super has the lowest safety factor. Only in two cases is the permanent deformation load incorrectly lower than yield load: mid IB and stern IB. This wrong permanent deformation load is obtained because unloading portion of the curve was not available. And slope of the curve that was used was much higher.

For Agulhass, the same trends can be observed as for Uikku. However, margin of safety against yield and permanent deformation load is much higher for most cases.

Table 7.1. Comparison of capacity levels for Uikku finite element models (different ice classes and load height)

Uikku	BOW			MID			STERN		
	IAS	IA	IB	IAS	IA	IB	IAS	IA	IB
Load for permanent deformation (kN)	958	770	664	480	400	191	360	217	84.8
Yield limit (kN)	598.8	506.7	437.6	345.5	184.26	207.29	253.3	184.26	138.19
p3h load (kN)	725.20	551.25	406.00	424.98	362.21	278.69	309.93	238.35	192.50
Design pressure (MPa)	3.44	2.92	2.41	1.78	1.51	1.25	1.34	1.16	0.8
patch height	0.35	0.3	0.25	0.35	0.3	0.25	0.35	0.3	0.25
patch width	0.35	0.35	0.35	0.343	0.343	0.343	0.35	0.35	0.35
A	0.12	0.11	0.09	0.12	0.10	0.09	0.12	0.11	0.09
p3h pressure (MPa)	5.92	5.25	4.64	3.54	3.52	3.25	2.53	2.27	2.2
Design load [kN]	421.40	306.60	210.88	213.69	155.38	107.19	164.15	121.80	70.00
% (design load / Permanent def.Limit)	44%	40%	32%	45%	39%	56%	46%	56%	83%
% (design load / Yield Limit)	70%	61%	48%	62%	84%	52%	65%	66%	51%

Table 7.2. Comparison of limit states for Agulhass finite element models (different ice classes and load height)

Agulhass	BOW			MID			STERN		
	IAS	IA	IB	IAS	IA	IB	IAS	IA	IB
Load for permanent deformation (kN)	1122	870	700	883	600	400	573	405	244
Yield limit (kN)	598.8	596.7	501.23	739.93	501.23	358.02	525.11	381.89	286.42
p3h load (kN)	925.40	684.00	502.00	691.60	559.20	461.00	560.00	460.80	395.00
Design pressure (MPa)	3.15	2.68	2.2	1.69	1.44	1.19	1.27	1.1	0.76
patch height	0.35	0.3	0.25	0.35	0.3	0.25	0.35	0.3	0.25
patch width	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4
A	0.14	0.12	0.10	0.14	0.12	0.10	0.14	0.12	0.10
p3h pressure (MPa)	6.61	5.7	5.02	4.94	4.66	4.61	4	3.84	3.95
Design load [kN]	441.00	321.60	220.00	236.60	172.80	119.00	177.80	132.00	76.00
% (design load / Permanent def.Limit)	39%	37%	31%	27%	29%	30%	31%	33%	31%
% (design load / Yield Limit)	74%	54%	44%	32%	34%	33%	34%	35%	27%

7.2 Case study II: The effect of load height on analysis of NUPP

The discussion in this section is also focused on the response of ship structures when loaded with a NUPP. Unlike in previous section, here the load height was kept constant for all the models - 0.35 m. The load vs. displacement data for the models have been shown in Figures 7.13 and 7.14. Three different hull locations (bow, mid-body and stern) of two ships, MT Uikku and SA Agulhass II, have been used and for three different ice classes IAS, IA and IB. This has been done to test the effect of ice-strengthening on the structural response. In total, there are 15 models that are analysed in this study.

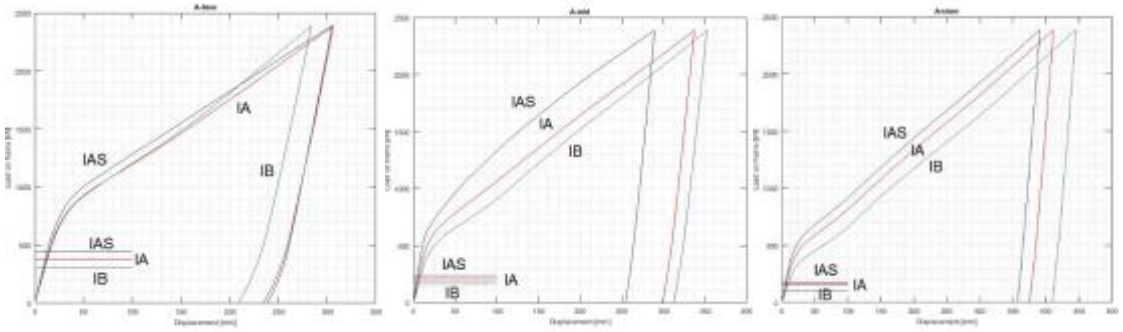

Figure 7.13. Load on frame (kN)-displacement (mm) plots of Agulhass bow (left), mid-ship (middle) and stern (right) for three ice classes: IAS (blue), IA (red) and IB (green). The constant FSICR design load is also plotted for each ice class.

Figures 7.13 and 7.14 show that for all cases of Agulhass and Uikku ships, a lower ice class doesn't affect the deformations when the load is applied over the patch of same size. This is evident from the similarity of the curves plotted when the ice-classes are reduced. However, there is a slight deviation for such behavior in the mid-ship and stern locations, where the deformations increase for low ice-strengthened structures when the same load is applied.

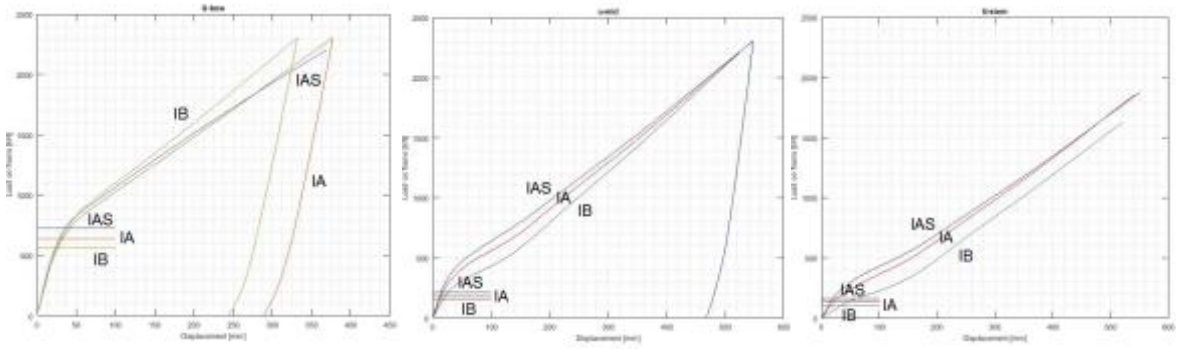

Figure 7.14. Load on frame (kN)-displacement (mm) plots of Uikku bow (left), mid-ship (middle) and stern (right) for three ice classes: IAS (blue), IA (red) and IB (green). The constant FSICR design load is also plotted for each ice class.

For the case of Uikku, the behaviour in the different locations is somewhat more consistent then Agulhass and a lower ice class affects the resulting deformations less. Three capacity levels for the ship MT Uikku and for three ice classes have been summarized in Tables 7.3 and 7.4.

Table 7.3. Comparison of capacity levels for Uikku finite element models (different ice classes)

	IAS	IA	IB
Load for permanent deformation (kN)	967.3	921.3	990.3
Yield limit (kN)	598.81	598.81	621.84
p3h load (kN)	725.20	643.13	568.40
Design pressure	3.44	2.92	2.41
patch height	0.35	0.35	0.35
patch width	0.35	0.35	0.35
A	0.12	0.12	0.12
p3h pressure (MPa)	5.92	5.25	4.64
Design load [kN]	421.40	357.70	295.23
% (design load / Permanent def.Limit)	44%	39%	30%
% (design load / Yield Limit)	70%	60%	47%

It can be observed from Rows 1-3 that two capacity levels, load for permanent deformation and yield load, remain mostly constant across different ice classes. The third capacity state, 3-hinge pressure, still decreases from IAS to IB but not substantially. As a constant load height has been used for all these cases, the load patch area A remains the same. Hence, the pressure is less concentrated in the analysis of IA and IB compared with analysis in previous section. However, when the safety margins of Case Study I and Case Study II are compared, it is not seen to be affected.

Table 7.4. Comparison of limit states for Agulhass finite element models (different ice classes)

	IAS	IA	IB
Load for permanent deformation (kN)	1146	1050	1050
Yield limit (kN)	716.06	692.19	716.06
p3h load (kN)	925.40	798.00	702.80
Design pressure	3.15	2.68	2.2
patch height	0.35	0.35	0.35
patch width	0.4	0.4	0.4
A	0.14	0.14	0.14
p3h pressure (MPa)	6.61	5.7	5.02
Design load [kN]	441.00	375.20	308.00
% (design load / Permanent def.Limit)	38%	36%	29%
% (design load / Yield Limit)	62%	54%	43%

In Table 7.4, the same numbers for SA Agulhass II ship have been documented for three ice classes. The trends in this table is in-line with that of Uikku (Table 7.3). Observations show that

there are no marked changes in the first two capacity levels. Furthermore, the safety margins are not affected when Table 7.4 is compared with the Table 7.2 where load height was changing.

7.3 Case study III: Analysis with uniform FSICR design load

In this section, the cases are discussed where a real FSICR pressure patch is modelled, with a single frame in each of the models being loaded with a uniform FSICR design load. The data from the single frame has been plotted in the form of a load-displacement curve, as shown in Figures 7.15-17. In these figures, the response due to NUPP is compared with response due to uniform FSICR design load.

It is to be noted that similar to Case I, different ice classes have been loaded with different load heights to test the effect of both on 15 different finite element models. This leads to a load patch with more concentrated area for lower ice classes, effects of which can be seen in the figures discussed in this chapter.

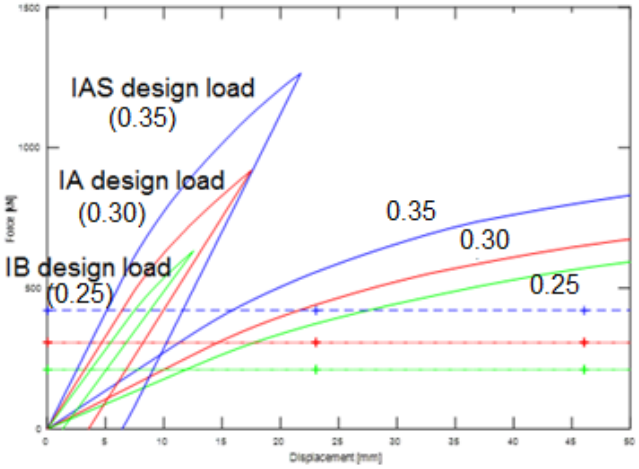

Figure 7.15. Load - displacement plots of Uikku bow. NUPP response compared with a uniform FSICR design load applied on single frame. Horizontal line depicts the magnitude of the FSICR design load.

It can be observed from the Figure 7.15 that the design load, compliant with the Finnish-Swedish Ice Class Rules and imposed on a single frame, yields much stiffer response compared with decreasing load function. Hence, the deformations are also much lower for FSICR design load than the ones caused by NUPP (when compared at the same load level). The FSICR design load value has also been plotted in the Figure 7.15 for reference purposes.

These results show that FSICR patch yields much higher structural stiffness compared with the NUPP application over the webframe spacing. This makes the NUPP approach less conservative compared with present design approach.

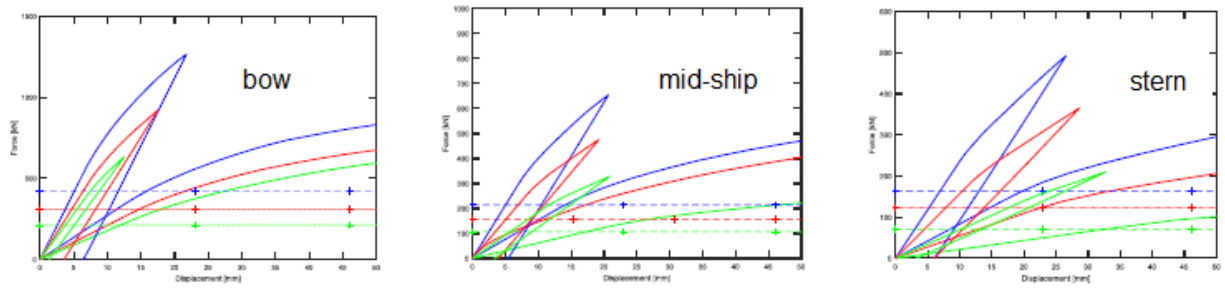

Figure 7.16 Load - displacement plots for Uikku using two different methods: NUPP and FSICR. IAS with load height of 0.35 m (blue), IA 0.30 m (red) and IB 0.25 m (green). Horizontal lines depicts the magnitude of the FSICR design load for each ice class.

When the Force-displacement plots for constant FSICR design load are compared for different locations in the ship hull (Figure 7.16), it is found that the magnitude of load decreases from bow to stern. Consistent with previous observations about FSICR design load, the response caused in each of these cases are also noticeably much stiffer than the cases where decreasing load function has been applied on the webframe.

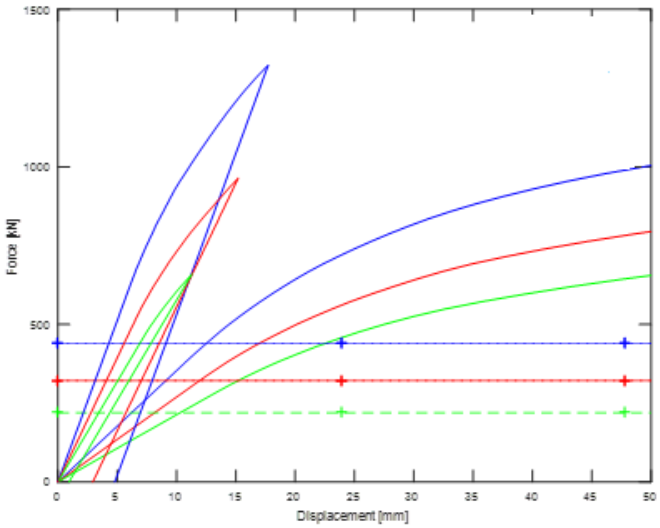

Figure 7.17. Load - displacement plots for Agulhass using two different methods: NUPP and FSICR. Horizontal line depicts the magnitude of the FSICR design load. Load plotted for single frame.

When the same plot is obtained for Agulhass bow, the behavior (Figure 7.17) is similar to that of Uikku's bow. When FSICR design load is applied to a single frame, it also results in a much stiffer response than that resulting from NUPP.

7.4 Case Study IV: Decreasing load function using ice-class specific engine power

In the analyses conducted in Chapters 7.1 and 7.2, engine power was assumed to be equivalent to ''as built'' power, and was taken to be constant for all ice classes IAS, IA and IB. However, according to design rules the minimum required engine power is also an ice class dependent parameter, which affects the structural scantlings. To take this into account in this section, the minimum required engine power for each ice class has been used, and the resulting scantlings were re-calculated using Matlab. Using these scantling dimensions, new Finite Element models were generated with the parametric module in Abaqus/CAE. After performing the same simulations as done in previous sections, the results have been compared with Case I to show the effect of engine power on the structural response.

Similar to Case Study I and III, different load heights have been used when applying the NUPP on models of different ice classes. Thus, lower ice classes are subjected to pressure with more concentrated area.

The minimum engine power requirements have been taken from the FSICR handbook (see Appendix 4 for the equations). As expected, the calculated minimum required engine power is lower for ice classes of IA and IB as compared to IA Super, which translates to lower applied pressure on the framing and plating. The force-displacement plots for the new models are shown in Figures 7.18-7.19 for the case of Agulhass bow, midship and stern for two different ice classes IA and IB. These plots aim to show a comparison between the plots using constant engine power (Case Study I) and the ice-class specific engine power, ultimately highlighting the effect of engine power on the analysis results.

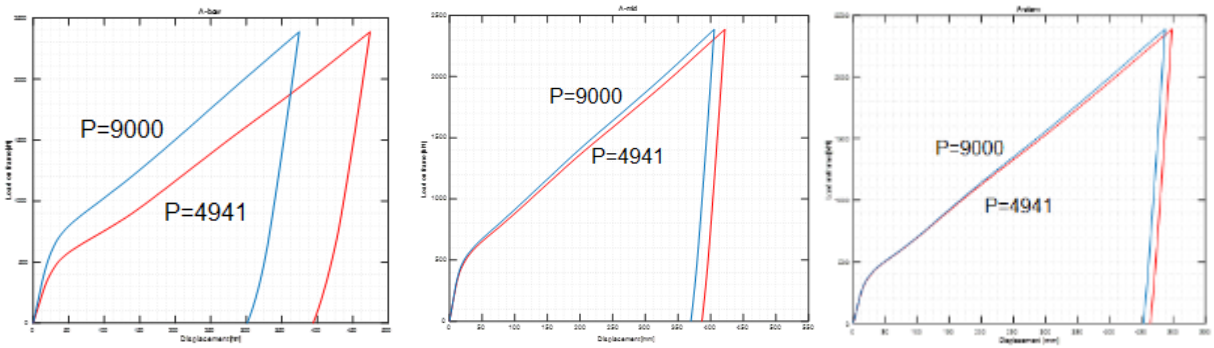

Figure 7.18. Load-displacement curves for the constant engine power (blue) vs the ice-class specific engine power (red). The cases demonstrated are Agulhass bow (left), mid (center) and stern (right). The results are plotted for ice class IA and load height of 0.30 m. Units of engine power P are in kW.

The curves in Figure 7.18 demonstrate that when the ice class-specific engine power is used displacements increase notably for Agulhass bow. Conversely, engine power does not influence the response at midship and stern. In other words, engine power does not affect the scantlings of midship and stern according to FSICR as it relates to the momentum of ship when it hits the ice field coming from open-water. During this event loads are carried by the bow region that is properly recognized by the current rules. A similar behaviour for all three structural locations is also observed if the respective curves are obtained for ice class IB, as shown in Figure 7.19.

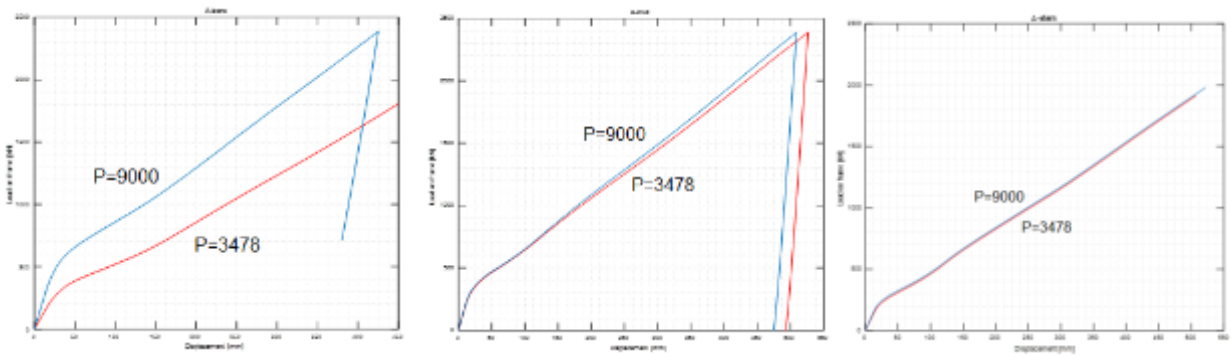

Figure 7.19. Load-displacement curves for the constant engine power (blue) vs the ice-class specific engine power (red). The cases demonstrated are Agulhass bow (left), mid (center) and stern (right). The results are plotted for ice class IB and load height of 0.25 m. Units of engine power P are in kW.

When the comparison is made between the same set of curves obtained for Uikku ship's locations, the behaviour observed is similar to the cases for Agulhass. Consistent with the latter's observations, the curves for Uikku in Figures 7.20-4.21 demonstrate a notable difference in

displacement for bow location, whereas there is only marginal (or negligible) increase in deformations for the mid and stern locations.

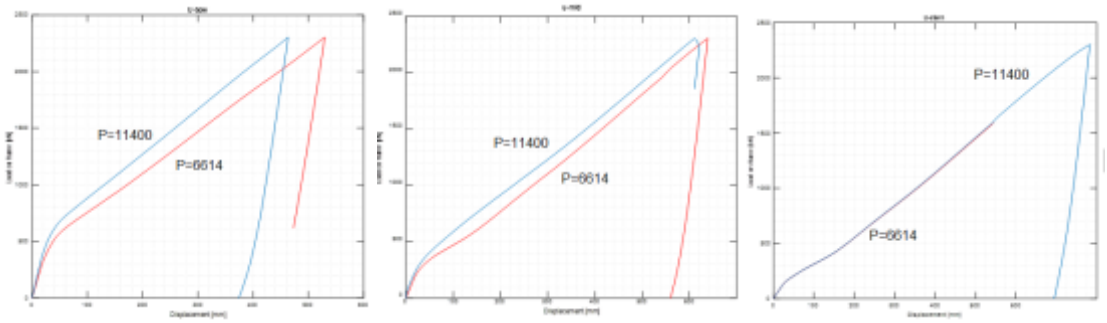

Figure 7.20. Comparison of Load on frame (kN)-displacement (mm) curves for the constant engine power (blue) vs the ice-class specific engine power (red). The cases demonstrated are Uikku bow (left), mid (center) and stern (right). The results are plotted for ice class IA and load height 0.30. Units of engine power P are in kW.

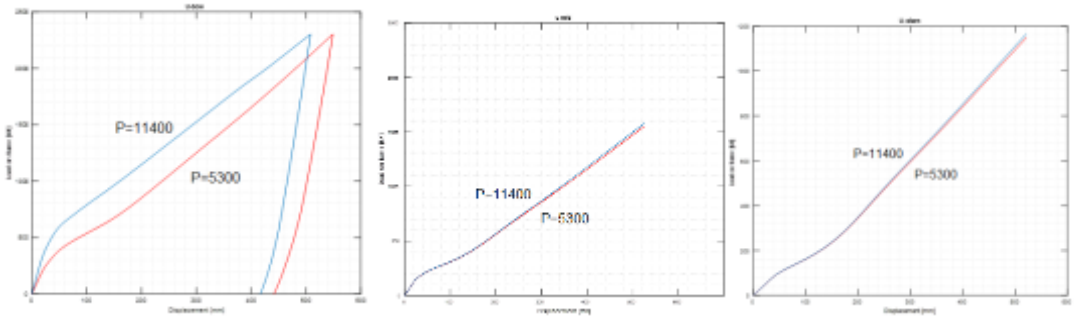

Figure 7.21. Comparison of Load on frame (kN)-displacement (mm) curves for the constant engine power (blue) vs the ice-class specific engine power (red). The cases demonstrated are Uikku bow (left), mid (center) and stern (right). The results are plotted for ice class IB and load height 0.25. Units of engine power P are in kW.

To build on the above findings, three capacity levels and safety margins of those against design point have been investigated similarly as in previous section, see Tables 7.1 and 7.2 that summarise safety margins obtained from different models of MT Uikku and SA Agulhass II ships.

Table 7.5. Comparison of limit states for Uikku finite element models (engine power specific to each ice class)

Uikku	BOW			MID			STERN		
	IAS	IA	IB	IAS	IA	IB	IAS	IA	IB
Load for permanent deformation (kN	990,4	667,9	437,6	506,7	345,5	207,3	299,4	184,3	69,1
Yield limit (kN)	598,81	437,62	299,41	345,47	276,39	207,29	253,34	184,26	138,19
p3h load (kN)	725,2	551,25	354,375	424,977	362,208	277,83	309,925	238,35	191,625
Design pressure	3,39	2,82	2,19	1,76	1,48	1,18	1,32	1,13	0,76
p3h pressure (MPa)	5,92	5,25	4,05	3,54	3,52	3,24	2,53	2,27	2,19
Design load [kN]	415,275	296,1	191,625	211,288	152,292	101,185	161,7	118,65	66,5
% (design load / Permanent def.Limi	42 %	44 %	44 %	42 %	44 %	49 %	54 %	64 %	96 %
% (design load / Yield Limit)	69 %	68 %	64 %	61 %	55 %	49 %	64 %	64 %	48 %

In Tables 7.5 and 7.6, it has been observed that all three capacity levels follow the same trend as in Case Study I. That is, the capacity decrease from IAS to IB, and from bow to stern (stern being the lowest). When comparing the yield point of this case with that of Case Study I (Table 7.1), there is a similarity in these values for IA and IB ice classes of mid-ship and stern locations. This is consistent with the findings from Figures 7.19 and 7.21 which show the plots of the two cases overlapping for mid-ship and stern locations. Moreover, these values are observed to be decreasing for the bow location, which also fits with the behavior seen in the Figures 7.19 and 7.21.

Table 7.6. Comparison of limit states for Agulhass finite element models (engine power specific to each ice class)

Agulhass	BOW			MID			STERN		
	IAS	IA	IB	IAS	IA	IB	IAS	IA	IB
Load for permanent deformation	1050	644,4	405,8	811,5	620,6	429,6	644,4	477,4	286,4
Yield limit	644,45	453,49	334,15	668,32	501,23	358,02	525,11	381,89	262,55
p3h	5,47	4,81	4,36	4,47	4,66	4,61	4	3,84	3,95
Design pressure	3,02	2,27	1,71	1,66	1,33	1,05	1,25	1,02	0,68
Design load [kN]	422,8	272,4	171	232,4	159,6	105	175	122,4	68
p3h load	765,8	577,2	436	625,8	559,2	461	560	460,8	395
% (design load / Permanent def.Li	40 %	42 %	42 %	29 %	26 %	24 %	27 %	26 %	24 %
% (design load / Yield Limit)	66 %	60 %	51 %	35 %	32 %	29 %	33 %	32 %	26 %

From Table 7.6, it can be observed that when the values of first two capacity levels (row 1 and 2) are compared with that from Case Study I (Table 7.2), there is a similarity/marginal increase in these values for IA and IB ice classes of mid-ship and stern locations. This is consistent with the findings from Figures 7.18 and 7.20 which show the plots of the two cases overlapping for mid-ship and stern locations. In addition, these values are observed to be decreasing for the bow location, which also fits with the behavior seen in the Figures 7.18 and 7.20. The third capacity state, 3-hinge pressure, is seen to be lower as compared to Case Study I for all the 15 models summarized in Tables 7.5 and 7.6.

8 Conclusions

This thesis has investigated the critical loading scenarios of multiple ice-strengthened ship structures when subjected to different load configurations. The structural performance for 15 different finite element models has been evaluated using Non-linear Finite Element Analyses. A non-uniform pressure patch (NUPP) was used to investigate the capacity of structures as well as to determine the safety margins against different capacity levels with respect to FSICR design load. Four different case studies were discussed focusing on the response of finite element models when subjected to different loading scenarios. These case studies also investigated the effect of load height and engine power on the structural response.

8.1. Case Study I: Response to Non-uniform pressure patch (NUPP)

The results of the *Case Study I* showed that for Agulhass and Uikku ships, a lower ice class and load height led to larger deformations when the same ice load is applied. The design point according to FSICR is reaching yield, but when the yield load was compared with the design load, structures had considerable reserve capacity against yield. This margin was lowest for IA Super indicating that the design complies well with the implicit design point of yield, but for other classes (IA and IB) the safety margin against yield was almost a factor of two. As seen in the state-of-the-art (*Chapter 2.2*), the additional reserve capacity for ice-strengthened structures was not fully clarified earlier. This goal of the thesis was achieved in *Case Study I* and two other case studies in which the reserve capacity was computed for both ships built with each of the defined ice classes.

In terms of the hull location, Uikku mid-ship and stern locations undergo much higher deformations than the corresponding bow location when subjected to the same loading scenario. To draw some conclusions based on the contour plots obtained at various stages of the loading cycle, three further cases were designed and their results examined. Results validated that for the Uikku ship, the ice-strengthened properties of its bow location are more than that of the mid-ship and stern locations. Further, mid-ship location required a higher load than the stern location to reach the yield limit. Unlike for Uikku, bow and mid-ship locations of Agulhass show similar deformations when subjected to the same ice load. However, in the corresponding stern location, increased displacements can be observed when the same ice load is applied.

Comparison of capacity levels showed that the yield point and load for permanent deformation are highest for a IA Super ice class that is consistent with the design approach. These capacities decrease for weaker ice classes.

8.2. Case Study II: Effect of load height on analysis of NUPP

In the *Case Study II*, the load height is kept constant across models built with different ice classes. Interestingly, the difference between ice classes almost completely diminished, although higher ice classes should be stronger than lower ice classes. Consequently, the load height has important effect on the response that should be addressed in future studies.

8.3. Case Study III: Analysis with uniform FSICR design load

In the *Case Study III*, a single frame in each ship structure has been loaded with a constant load equivalent to FSICR design load. It was observed that this load yields much stiffer response compared with NUPP. Therefore, if the FSICR design approach had been used in the Case study I, the safety margins against different capacity levels would have been considerably higher. There are two possible ways to interpret this conclusion. First, the present design approach is extremely conservative and safe, even against point of yield. Second, the current design pressure patch approach simplifies the actual process to a degree that makes structures look safer than they actually are. It is likely that the latter is more probable than the former, which also justifies the deployment of less conservative non-uniform pressure patch (NUPP). This is in agreement with the state-of–the-art (*Chapter 2.1*) which stated that the FSICR design approach leads to extremely conservative results in comparison to more complex load shapes such as NUPP.

8.4. Case Study IV: Decreasing load function using ice-class specific engine power

Finally, in *Case Study IV* the effect of engine power on the response was studied. Results for IA and IB ice classes demonstrated that reduced engine power leads to a noticeable increase in displacements at bow. For mid and stern locations, however, the reduced engine power caused no substantial effect on the deformations. This demonstrates how engine power intertwines with the bow design to account for the extra load bow carries due to the increased speeds when navigating in ice.

8.5. Limitations and future steps

Due to time limitations, the finite element models used for the structural analysis were simplified to some extent. Future investigations could focus on verification studies of the size of the model and parameters, such as the element mesh size, element types, and boundary conditions. It has to be pointed out that the modelling approach used in investigating the nature of the ice load had some uncertainty because the whole pressure load patch had to be scaled up to inflict more deformations on the structure.

References

Riska, Kaj, and Jorma Kämäräinen. "A review of ice loading and the evolution of the finnish-swedish ice class rules." *Proceedings of the SNAME Annual Meeting and Expo. November*. 2011.

Riska, K., 2014. Design point in ice class rules. Finnish Transport Safety Agency & Swedish Maritime Administration.

Kujala, Pentti. Damage Statistics of Ice-strengthened Ships in the Baltic-Sea, 1984-1987. Finnish Government Print. Centre, 1991.

Hänninen, Samuli. "Design ice load level in the Baltic Sea." Proceedings of the International Conference on Port and Ocean Engineering Under Arctic Conditions. 2003.

Lyngra, Nora Helen Lund. "Analysis of Ice-Induced Damages to a Cargo Carrier and Implications wrt. Rule Requirements." (2014).

Kurmiste, Andres. "Analysis of structural safety of ice-going vessels in the Arctic and Antarctic." (2016).

Kaldasaun, Jorma. "Risk-Based Approach for Structural Design of Ice- Strengthened Vessels Navigating in the Baltic Sea." (2010).

Suominen, Mikko, and Pentti Kujala. "The Measured Line Load As A Function Of The Load Length In The Antarctic Waters." *Proceedings of the 23rd International Conference on Port and Ocean Engineering under Arctic Conditions*. 2015.

Maritime Safety Regulation by Transport Safety Agency (TRAFI). 2010.

Quinton, Bruce, Claude G. Daley, and Robert E. Gagnon. "Effect of moving ice loads on the plastic capacity of a ship's structure." *Proceedings of the 9th International Conference and Exhibition on Performance of Ships and Structures in Ice, Anchorage, Alaska, USA*. 2010.

Erceg, Boris, et al. "A response comparison of a stiffened panel subjected to rule-based and measured ice loads." *ASME 2014 33rd International Conference on Ocean, Offshore and Arctic Engineering*. American Society of Mechanical Engineers, 2014.

Kõrgesaar, Mihkel, and Pentti Kujala. “Modeling of ice loads with pressure patches in the analysis of ship structures.” *Proceedings of the Arctic Technology Conference, St. John’s, Newfoundland and Labrador.* 2016. (under review)

Kujula, P. "On the statistics of ice loads on ship hull in the Baltic." (1994).

Abraham, J., and C. J. Daley. “Load Sharing In A Grillage Subject To Ice Loading.” *International Conference on Ship and Offshore Technology Ice Class Vessels. Busan Korea.* 2009.

Kujala, Pentti, and Jukka Vuorio. *Results and statistical analysis of ice load measurements on board icebreaker Sisu in winters 1979 to 1985*. Sjöfartsverket, 1986.

Sukselainen, J., and T. Nyman. The maximum plate stress and ice pressure in 1987 obtained the highest values of the whole long-term measurements 1983–87. *Research Report No. LAI – 331E/88. Ship Laboratory, TKK, Finland. (unpublished)* 1988.

Requirements Concerning Polar Class by International Association of Classification Societies (IACS). 2011. IACS Req. 2006/Rev.02, 2010.

Wang, G., Shewen Liu, and Kaj Riska. "Recent Advances in Structural Design of Ice-Strengthened Vessels." *Proceedings of the Seventh International Conference and Exhibition on Performance of Ships and Structures in Ice*. 2006.

Wang, G., et al. "Rationalizing the design of ice strengthened side structures." *Proceedings of Maritime Transportation and Exploitation of Ocean and Coast Resources, Lisbon, Portugal* (2005).

Taylor, Rocky S., et al. "Local design pressures for structures in ice: analysis of full-scale data." Journal of Offshore Mechanics and Arctic Engineering 132.3 (2010): 031502.

Kujala, Pentti, and Sankar Arughadhoss. "Statistical analysis of ice crushing pressures on a ship's hull during hull–ice interaction." *Cold Regions Science and Technology* 70 (2012): 1-11.

Systemès, Dassault. "ABAQUS Version 6.14, user documentation." *Dassault Systemes, Providence, RI* (2014).

Reddy, Junuthula Narasimha. *An Introduction to Nonlinear Finite Element Analysis: with applications to heat transfer, fluid mechanics, and solid mechanics*. OUP Oxford, 2014.

Bathe, Klaus-Jürgen. *Finite element procedures*. Klaus-Jurgen Bathe, 2006.

List of appendices

The appendices that were referred to in this thesis have been listed in this section.

Appendix 1. Calculation of ship Scantlings using FSICR equations. 15 pages.

Appendix 2. Parametric model scripts (for the case of Uikku bow IAS). 28 pages.

Appendix 3. Calculation of ship scantlings using FSICR equations. 2 pages.

Appendix 1: Calculation of ship scantlings using FSICR equations

The scantlings used for generating the parametric model of the ships can be obtained using Finnish-Swedish Ice Class Rules equations (Maritime Safety Regulation, 2010). These equations are computationally evaluated with Matlab using the ship input dimensions to render the required scantlings. The ice pressure, shell plating, stringers and web frames are determined by the formulae below:

1.1. Ice pressure

The design ice pressure is calculated by:

$$p = c_d . c_p . c_a . p_0 \text{ MPa}$$

where

c_d is a factor accounting for influence of the ship's size and engine output

c_p is a factor accounting for the probability that the design ice pressure is being applied at a certain location of the ship hull (ice-class specific)

c_a is a factor accounting for the probability that the entire length of the patch area will be subjected to the pressure at a particular instance in time

p_o is nominal ice pressure with value 5.6 MPa

1.2. Design of shell plating

For transverse framing the thickness of the shell plating, the following formula is used:

$$t = 667\, s \sqrt{\frac{f_1 \cdot p_{PL}}{\sigma_y}} + t_c \,[\text{mm}]$$

On the other hand, the thickness of the shell plating for longitudinal framing is:

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

1.3. Design of frames

1.3.1. Section modulus and shear area

For calculating the section modulus of a transverse frame, the following formula is used:

$$Z = \frac{p \cdot s \cdot h \cdot l}{m_t \cdot \sigma_y} 10^6 \text{ [cm}^3\text{]},$$

For determining the effective shear area, the following can be used:

$$A = \frac{\sqrt{3} \cdot f_3 \cdot p \cdot h \cdot s}{2\sigma_y} 10^4 \text{ [cm}^2\text{]},$$

where

p is the ice pressure [in MPa]

s is the frame spacing [in m]

h is the height of load area [in m]

l is the span of the frame [in m]

f_3 is a factor that takes into account the maximum shear force vs. load location and the shear stress distribution, $f_3 = 1.2$

σ_y is the yield stress [in N/mm^2]

1.3.2. Web thickness of framing

Web thickness of the frames can be determining by taking the maxima of the following four values:

- $\frac{h_w\sqrt{\sigma_y}}{C}$ (hw is web height, $C = 282$ for flat bars & $C = 805$ for profiles)
- 2.5 % of frame spacing (transverse frames);

- 1/2 of shell plating's net thickness, $t - t_c$ (For calculating the web thickness of frames)
- 9 mm.

1.4 Stringer design

If a stringer is situated within the ice belt, its section modulus can be calculated as follows:

$$Z = \frac{f_6 \cdot f_7 \cdot p \cdot h \cdot l^2}{m \cdot \sigma_y} 10^6 \quad [\text{cm}^3].$$

It's effective shear area can be calculated as:

$$A = \frac{\sqrt{3} \cdot f_6 \cdot f_7 \cdot f_8 \cdot p \cdot h \cdot l}{2 \cdot \sigma_y} 10^4 \quad [\text{cm}^2],$$

where

p is the ice pressure [in MPa]

h is height of the load area [in m]

Please note that $p.h$ cannot be less than 0.15.

l is the stringer span [in m]

m is a factor that takes into account the boundary condition

f_6 is a factor accounting for the distribution of load to transverse frames (taken as 0.9)

f_7 is safety factor for stringers (taken as 1.8)

f_8 is a factor accounting for the maximum shear force vs. load location and the shear stress distribution; $f_8 = 1.2$

σ_y is the yield stress [in N/mm^2]

1.5 Webframe design

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

$$F = f_{12} \,.\, p \,.\, h \,.\, S \,[\text{MN}]$$

where

p is the ice pressure [in MPa]

h is height of the load area [in m]

Please note that $p.h$ cannot be less than 0.15.

S is the web frame spacing [in m]

f_{12} is factor of safety for web frames (taken as 1.8)

The shear area for web frames can be calculated by the following equation:

$$A = \frac{\sqrt{3} \cdot \alpha \cdot f_{13} \cdot Q \cdot 10^4}{\sigma_y} \quad [\text{cm}^2],$$

where

Q is the maximum shear force as calculated under the ice load F

f_{13} is a factor accounting for the shear force distribution, $f_{13} = 1.1$

σ_y is the material yield stress.

For the calculation of the maximum calculated shear force (Q), the webframe was assumed to be loaded where the stringers are located. In total, there are 4 stringers along the web height. For the case of Agulhass mid-ship, the spacing between each is 2.065m, whereas for Kemira and Uikku the spacing for mid-ship is 3.2m and 2.1 m respectively. In each stringer location, the ice load F is acting on the webframe. For calculating the shear force distribution in the web, the webframe was assumed to be a beam with the stringer loads along its height, as shown in Figure A1 below.

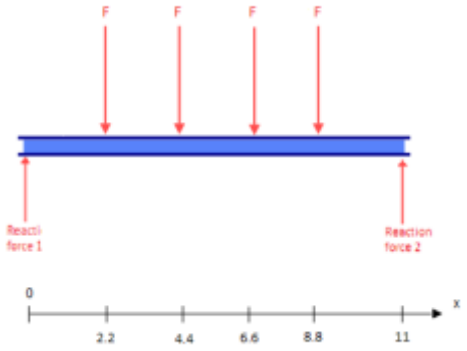

Figure A1. Point forces (due to stringers) acting on the webframe (assumed to be a beam) of Agulhass mid-ship

According to this beam analysis, maximum calculated shear force was computed to be 2F, as shown in the Shear Force Diagram below. This value of shear force (Q=2F) was used in the shear area calculations. After testing the effect of shear force on the resulting section modulus, it was found to have very little effect. Hence, Q=F/2 has been assumed.

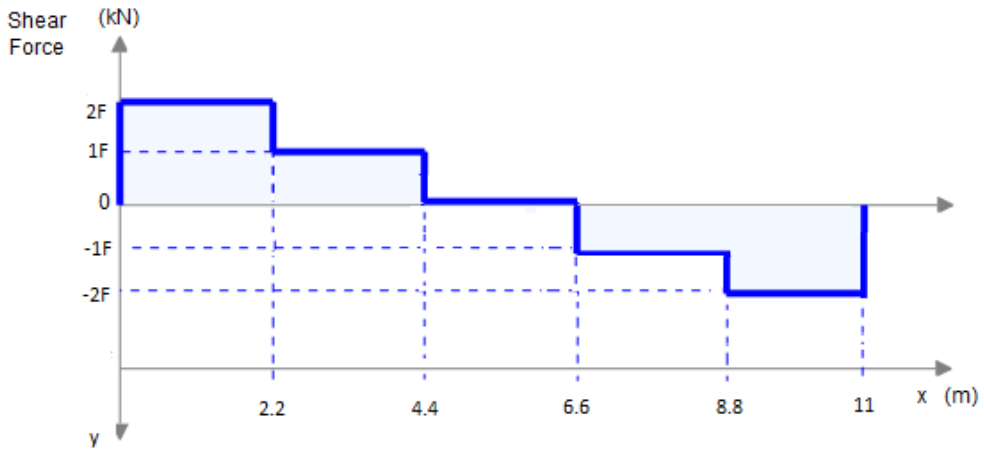

Figure A2. Shear force diagram (SFD) along the beam (webframe) Agulhass mid-ship

Shear modulus for the webframe can be computed using the equation below:

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

where

M is the maximum calculated bending moment under the ice loading F, which can be calculated as $M = 0.193.F.l$

A is the required shear area

Aa is the actual area of cross section for the webframe, $Aa = Af + Aw$

Appendix 2: Parametric model scripts (for the case of Uikku bow IA Super)

Appendices 2.1 to 2.3 lists the three python scripts that have been used for generating the parametric finite element models using the parametric module in Abaqus/CAE. When the main script (Appendix 2.1) is compiled, it calls the Variables script (Appendix 2.2) and Grillage script (Appendix 2.3) to eventually create the parametric models for each of case studies in this thesis.

2.1. Main_script.py

```
# General definations
=============================================================================
from abaqus import *
from abaqusConstants import *
from caeModules import *
import testUtils
import numpy
from operator import truediv
testUtils.setBackwardCompatibility()
# Helps making the python script for creating sets, surfaces, selecting regions etc. more readable
session.journalOptions.setValues(replayGeometry=COORDINATE,
recoverGeometry=COORDINATE)
# Create viewport
myViewport = session.Viewport(name='Viewport: 1', origin=(0.0, 0.0), width=100, height=100)
myViewport.makeCurrent()
myViewport.maximize()
# Run scripts
execfile('C:/Users/ghoshs2/Dropbox/Ghosh/Analyses/uikku/bow/1AS/center/Variables_UIKKU
_1AS.py') #newchange
execfile('C:/Users/ghoshs2/Dropbox/Ghosh/Analyses/uikku/bow/1AS/center/TRAFI_Grillage_S
cript.py') #newchange
# execfile('C:\work\TRAFI\scripting\Uikku_bow\meshrefinement.py') #newchange
```

2.2. Variables_script.py

==========

```
ship='Uikku'
iceclass='IAS'
location='bow'
implicit='yes'    # yes or no
loadpatchheight=0.35
pathclocation='wide' #plate or frame, if something else, no small patch is made
pressure=2.e+7       # this reference pressure can be kept the same for all analysis
if implicit=='yes':
   if pathclocation=='frame':
      ship='Uikku-web-imp-20'
   else:
      ship=ship+str('_')+iceclass+str('_')+location
else:
   ship='Uikku-bow-explicit'
# In explicit analysis lets define the load through user defined model
#changed: st_w is the webframe spacings, n_str is the number of shell panels
n_str=5
st_w=[2.8,2.8,2.8,2.8,2.8] #newchange
# Plate and frames
nx=7; n=[nx,nx,nx,nx,nx]        # MKmod 21.07.16 - this has to be consistent with spacing
spacing = [x / (n[0]+1) for x in st_w]
# spacing = [0.35, 0.35, 0.35,0.35,0.35] #newchange
fh=[0.29,0.29,0.29,0.29,0.29]     # newchange: frame height
frf=0.04
frame_flange=[frf,frf,frf,frf,frf]    # newchanged: flange dimension (this is half from flange width)
tframe=[0.0095,0.0095,0.0095,0.0095,0.0095]  # newchange: thickness of frame
tframe2=[0.0145,0.0145,0.0145,0.0145,0.0145] #newchange: thickness of T flanges for frames, in
height direction
tp=0.0214
```

```
tshell=[tp,tp,tp,tp,tp] #newchange
# Stringers
girderH=0.63            #height
girder_flange=0.145/2.  #width (division by 2, because halfe of the T frames is modelled)
tgirder=0.018           #tw (web thickness)
tgirder2=0.021          #tf (flange thcikness)
# Webframes
webheight=0.9             #hw
webframe_flange=0.08/2.     #wf
tweb=0.015                #tw
tweb2=0.017               #ft
depthmp=5 # newchanged: this is the nummber_of_girders+1
depth=2.5 # newchange: girder spacing dimensions
depthx1=[depth, depth*2, depth*3, depth*4] #changed: girder spacings
depth1=depth*depthmp  # changed: length in frame direction - span of outer shell in the transverse direction
# Hard points
HPx1=[]; HPx2=[];
HPx1.append(0); HPx2.append(st_w[0]);
for i in range(n_str-1):
        HPx1.append(HPx1[i]+st_w[i])
        HPx2.append(HPx2[i]+st_w[i+1])
mesh_size=0.15
mesh_fine=0.05
sim_time=2.
Num_Int=200
# changed: girder locations
Gx1=[];
Gx1.append(0);
for i in range(depthmp-1):
        Gx1.append(depthx1[i])
# ============================================================
# refined mesh Patch coordinates (this is only for mesh refinement)
```

```
# ================================================================
margin=int(4.)*mesh_fine #this ensures that edges of the patch dont lie on frame (the multiplier in front can be changed for different models )
p_w=2.*st_w[0]+margin; p_h=depth+1.5   #patch width & patch height - these must be defined
# !!!!!!!!! modify these to move refined area
# pxz are used in the main script also to define sets , e.g. L389,
# where the assumption is that load is applied in the middle of the refined area (displacement output from middle)
#  pxz=[(sum(st_w)-p_w)/2.,(sum(st_w)+p_w)/2.,(depth1-p_h)/2.,(depth1+p_h)/2.]        # xmin, xmax zmin, zmax
x_patchshift=0.
z_patchshift=0.
pxz=[(sum(st_w)-p_w)/2.-x_patchshift,(sum(st_w)+p_w)/2.-x_patchshift,(depth1-p_h)/2.-z_patchshift,(depth1+p_h)/2.-z_patchshift]
# used as tolerance in findat command, this can be modified when script is used in mm for instance
tol=0.01
# ================================================================
# Material variables
# ================================================================
# Young's modulus
E = 2.0e+11
# Poission's ratio
nuu = 0.3
# Density
ro = 7850
# Fracture strain (depends on mesh element size - use failure_strain.exe)
ef = 0.35
# Stress-plastic strain curve
StressPStrain = ((280.0e+6, 0.000),(280.0e+6, 0.008),(280.0e+6, 0.010),(280.0e+6, 0.012),(280.0e+6, 0.013),(285.3e+6, 0.014),
   (288.5e+6, 0.015),(291.6e+6, 0.016),(302.6e+6, 0.020),(324.7e+6, 0.030),(356.1e+6, 0.050),(373.9e+6, 0.065),(388.9e+6, 0.080),
   (405.8e+6, 0.100),(427.0e+6, 0.130),(445.8e+6, 0.170),(455.2e+6, 0.190),(459.9e+6, 0.200),(464.6e+6, 0.210),(469.3e+6, 0.220),
```

(478.7e+6, 0.240),(488.1e+6, 0.260),(497.5e+6, 0.280),(506.9e+6, 0.300),(516.3e+6, 0.320),(525.7e+6, 0.340),(530.4e+6, 0.350),

(539.8e+6, 0.370),(549.2e+6, 0.390),(558.6e+6, 0.410),(568.0e+6, 0.430),(577.4e+6, 0.450),(586.8e+6, 0.470),(610.8e+6, 0.530),

(634.8e+6, 0.590),(658.8e+6, 0.650),(682.8e+6, 0.710),(706.8e+6, 0.770),(730.8e+6, 0.830),(754.8e+6, 0.890),(778.8e+6, 0.950),

(802.8e+6, 1.010),(826.8e+6, 1.070),(850.8e+6, 1.130),(874.8e+6, 1.190),(878.8e+6, 1.200))

#changed: this is not really required

Fracture strain (depends on mesh element size - use failure_strain.exe)

ef = 0.4060475

Stress-plastic strain curve

eps=[0,0.0025,0.01,0.02,0.0359,0.0387,0.0466,0.0576,0.0692,0.0805,0.0915,0.1026,0.1135,0.1283,0.1443,0.1607,0.1789,0.198,0.22,0.2453,0.276,0.3073,0.3482,0.3867,0.4354,0.4746,0.5185,0.5731,0.6387,0.701]

stress=eps*2

2.3. TRAFI_Grillage_Script.py

==========

```
# This script is for T profiles
import sketch
import part
# Create new model in model database
StrakeModel = mdb.Model(name='Model-1')
# Create new sketch for strake
Strake_Sketch = StrakeModel.ConstrainedSketch(name=('Sketch' + str(i)), sheetSize=3.0)
# Stored sketch geometries (Needed in linear patterning)
geo = Strake_Sketch.geometry
    # Create strakes and frames
for i in range(n_str):
    # Outer shell
    Strake_Sketch.Line(point1=(HPx1[i], 0.0), point2=(HPx2[i], 0))
    # first stiffener
    Strake_Sketch.Line(point1=(HPx1[i]+spacing[i], 0), point2=(HPx1[i]+spacing[i], fh[i]))
#     First L profile flange for the T flange
```

```
    Strake_Sketch.Line(point1=(HPx1[i]+spacing[i], fh[i]), point2=(HPx1[i]+spacing[i]+frame_flange[i], fh[i]))
#     Changed: Second L profile flange for the T flange
    Strake_Sketch.Line(point1=(HPx1[i]+spacing[i], fh[i]), point2=(HPx1[i]+spacing[i]-frame_flange[i], fh[i]))
# Make rest of the frames
    Strake_Sketch.linearPattern(geomList=(geo.findAt((HPx1[i]+spacing[i],fh[i]/2.)),), vertexList=(), number1=n[i], spacing1=spacing[i],
        angle1=0.0, number2=1, spacing2=0.3, angle2=90.0)
    Strake_Sketch.linearPattern(geomList=(geo.findAt((HPx1[i]+spacing[i]+0.001,fh[i])),), vertexList=(), number1=n[i], spacing1=spacing[i],
        angle1=0.0, number2=1, spacing2=0.3, angle2=90.0)
    Strake_Sketch.linearPattern(geomList=(geo.findAt((HPx1[i]+spacing[i]-0.001,fh[i])),), vertexList=(), number1=n[i], spacing1=spacing[i],
        angle1=0.0, number2=1, spacing2=0.3, angle2=90.0)
        # Create webframes
    if i > 0:
      Strake_Sketch.Line(point1=(HPx1[i], 0.0), point2=(HPx1[i], webheight))
                # Changed: T-webs
    if i > 0:
      Strake_Sketch.Line(point1=(HPx1[i], webheight), point2=(HPx1[i]+webframe_flange, webheight))
    if i > 0:
      Strake_Sketch.Line(point1=(HPx1[i], webheight), point2=(HPx1[i]-webframe_flange, webheight))
# Extruding
PanelPart=StrakeModel.Part(name='Panel', dimensionality=THREE_D, type=DEFORMABLE_BODY)
PanelPart.BaseShellExtrude(sketch=Strake_Sketch, depth=depth1)
# Girders #
# Create new sketch for girders
Girder_Sketch = StrakeModel.ConstrainedSketch(name=('girders'), sheetSize=3.0)
# Stored sketch geometries (Needed in linear patterning)
geox = Girder_Sketch.geometry
# Changed: new code for girder sketch
```

```
for i in range(depthmp-1):
   Girder_Sketch.Line(point1=(Gx1[i], 0), point2=(Gx1[i], girderH))
# changed:  T girder
for i in range(depthmp-1):
        Girder_Sketch.Line(point1=(Gx1[i],girderH), point2=(Gx1[i]+girder_flange, girderH))
        Girder_Sketch.Line(point1=(Gx1[i],girderH), point2=(Gx1[i]-girder_flange, girderH))
GirderPart=StrakeModel.Part(name='Girder', dimensionality=THREE_D,
type=DEFORMABLE_BODY)
GirderPart.BaseShellExtrude(sketch=Girder_Sketch, depth=sum(st_w))
# Create material
import material
# Create new material
Steel = StrakeModel.Material(name='Material-1')
# Material parameters
Steel.Elastic(table=((E,nuu),))
Steel.Plastic(table=StressPStrain)
Steel.Density(table=((ro,),))
Steel.DuctileDamageInitiation(table=((ef, 0.0, 0.0), ))
Steel.ductileDamageInitiation.DamageEvolution(
   type=DISPLACEMENT, table=((0.0, ), ))
# # Create section
==========================================================================
import section
for i in range(n_str):
        shell = StrakeModel.HomogeneousShellSection(('Sec_shell_' + str(i+1)),
      preIntegrate=OFF, material='Material-1', thicknessType=UNIFORM,
      thickness=tshell[i], thicknessField='', idealization=NO_IDEALIZATION,
      poissonDefinition=DEFAULT, thicknessModulus=None, temperature=GRADIENT,
      useDensity=OFF, integrationRule=GAUSS, numIntPts=5)

for i in range(n_str):
        frame = StrakeModel.HomogeneousShellSection(('Sec_fr_' + str(i+1)),
      preIntegrate=OFF, material='Material-1', thicknessType=UNIFORM,
```

```
        thickness=tframe[i], thicknessField='', idealization=NO_IDEALIZATION,
        poissonDefinition=DEFAULT, thicknessModulus=None, temperature=GRADIENT,
        useDensity=OFF, integrationRule=GAUSS, numIntPts=5)

for i in range(n_str):
            framefl = StrakeModel.HomogeneousShellSection(('Sec_flange' + str(i+1)),
        preIntegrate=OFF, material='Material-1', thicknessType=UNIFORM,
        thickness=tframe2[i], thicknessField='', idealization=NO_IDEALIZATION,
        poissonDefinition=DEFAULT, thicknessModulus=None, temperature=GRADIENT,
        useDensity=OFF, integrationRule=GAUSS, numIntPts=5)

web = StrakeModel.HomogeneousShellSection(name='Sec_web',
    preIntegrate=OFF, material='Material-1', thicknessType=UNIFORM,
    thickness=tweb, thicknessField='', idealization=NO_IDEALIZATION,
    poissonDefinition=DEFAULT, thicknessModulus=None, temperature=GRADIENT,
    useDensity=OFF, integrationRule=GAUSS, numIntPts=5)

webfl = StrakeModel.HomogeneousShellSection(name='Web_flange',
    preIntegrate=OFF, material='Material-1', thicknessType=UNIFORM,
    thickness=tweb2, thicknessField='', idealization=NO_IDEALIZATION,
    poissonDefinition=DEFAULT, thicknessModulus=None, temperature=GRADIENT,
    useDensity=OFF, integrationRule=GAUSS, numIntPts=5) #newchange

girder = StrakeModel.HomogeneousShellSection(name='Sec_girder',
    preIntegrate=OFF, material='Material-1', thicknessType=UNIFORM,
    thickness=tgirder, thicknessField='', idealization=NO_IDEALIZATION,
    poissonDefinition=DEFAULT, thicknessModulus=None, temperature=GRADIENT,
    useDensity=OFF, integrationRule=GAUSS, numIntPts=5)

girderfl = StrakeModel.HomogeneousShellSection(name='Girder_flange',
    preIntegrate=OFF, material='Material-1', thicknessType=UNIFORM,
    thickness=tgirder2, thicknessField='', idealization=NO_IDEALIZATION,
    poissonDefinition=DEFAULT, thicknessModulus=None, temperature=GRADIENT,
```

```
    useDensity=OFF, integrationRule=GAUSS, numIntPts=5) #newchange
=============================================================================
# Create face sets
=============================================================================
Panel_Faces = PanelPart.faces
# Outer shell face
for i in range(n_str):
        Outface=Panel_Faces.getByBoundingBox(HPx1[i]-0.1,0,-0.1,HPx2[i]+0.1,0,depth1+0.1)
        PanelPart.Set(faces=Outface,name=('shell_set' + str(i+1)))
# Frame face
for i in range(n_str):
    framex=[]
    flange1=[] #changed
    flange2=[] #changed
    for j in range(n[i]):
        framex.append(Panel_Faces.findAt(((HPx1[i]+(j+1)*spacing[i],0.05,0.3),)))
        flange1.append(Panel_Faces.findAt(((HPx1[i]+(j+1)*spacing[i]+0.001,fh[i],0.3),)))
#changed
        flange2.append(Panel_Faces.findAt(((HPx1[i]+(j+1)*spacing[i]-0.001,fh[i],0.3),)))
#changed
    PanelPart.Set(faces=framex,name=('frame_set' + str(i+1)))
    PanelPart.Set(faces=flange1,name=('Flange1_set' + str(i+1))) #changed
    PanelPart.Set(faces=flange2,name=('Flange2_set' + str(i+1))) #changed
# Web face
webx=[]
flangew1=[] #changed
flangew2=[] #changed
for i in range(n_str):
     if i > 0:
        webx.append(Panel_Faces.findAt(((HPx1[i],0.05,0.3),)))
        flangew1.append(Panel_Faces.findAt(((HPx1[i]+0.001,webheight,0.3),))) #changed
        flangew2.append(Panel_Faces.findAt(((HPx1[i]-0.001,webheight,0.3),))) #changed
PanelPart.Set(faces=webx,name='Webs')
```

```
PanelPart.Set(faces=flangew1,name='web_flange_1') #changed
PanelPart.Set(faces=flangew2,name='web_flange_2') #changed
# Changed: Girder face
GirderFaces=GirderPart.faces
girderx=[]
flangeg1=[] #changed
flangeg2=[] #changed
for i in range(depthmp-1):
    girderx.append(GirderFaces.findAt(((Gx1[i],0.05,0.3),))) #changed
    flangeg1.append(GirderFaces.findAt(((Gx1[i]+0.001,girderH,0.3),))) #changed
    flangeg2.append(GirderFaces.findAt(((Gx1[i]-0.001,girderH,0.3),))) #changed
GirderPart.Set(faces=girderx,name='Girders')
GirderPart.Set(faces=flangeg1, name='girder_flange_1') #changed
GirderPart.Set(faces=flangeg2, name='girder_flange_2') #changed
=========================================================================
# Assign sections
=========================================================================
import regionToolset
# Outer shell section assignment
for i in range(n_str):
  Outer_Shell_Region = PanelPart.sets[('shell_set' + str(i+1))]
  PanelPart.SectionAssignment(region=Outer_Shell_Region, sectionName=('Sec_shell_' +
str(i+1)), offset=0.0,
    offsetType=MIDDLE_SURFACE, offsetField='', thicknessAssignment=FROM_SECTION)
# frame assignment
for i in range(n_str):
  Frame_Region = PanelPart.sets[('frame_set' + str(i+1))]
  flange1_Region = PanelPart.sets[('Flange1_set' + str(i+1))] #changed
  flange2_Region = PanelPart.sets[('Flange2_set' + str(i+1))] #changed
  PanelPart.SectionAssignment(region=Frame_Region, sectionName=('Sec_fr_' + str(i+1)),
offset=0.0,
    offsetType=MIDDLE_SURFACE, offsetField='', thicknessAssignment=FROM_SECTION)
```

```
    PanelPart.SectionAssignment(region=flange1_Region, sectionName=('Sec_flange' + str(i+1)),
offset=0.0,
        offsetType=MIDDLE_SURFACE, offsetField='', thicknessAssignment=FROM_SECTION)
#changed
    PanelPart.SectionAssignment(region=flange2_Region, sectionName=('Sec_flange' + str(i+1)),
offset=0.0,
        offsetType=MIDDLE_SURFACE, offsetField='', thicknessAssignment=FROM_SECTION)
#changed
# WEb assignment
web_Region = PanelPart.sets['Webs']
flangew1_Region = PanelPart.sets['web_flange_1'] #changed
flangew2_Region = PanelPart.sets['web_flange_2'] #changed
PanelPart.SectionAssignment(region=web_Region, sectionName=('Sec_web'), offset=0.0,
    offsetType=MIDDLE_SURFACE, offsetField='', thicknessAssignment=FROM_SECTION)
PanelPart.SectionAssignment(region=flangew1_Region,          sectionName=('Web_flange'),
offset=0.0,
    offsetType=MIDDLE_SURFACE,  offsetField='',  thicknessAssignment=FROM_SECTION)
#newchange
PanelPart.SectionAssignment(region=flangew2_Region,          sectionName=('Web_flange'),
offset=0.0,
    offsetType=MIDDLE_SURFACE,  offsetField='',  thicknessAssignment=FROM_SECTION)
#newchange
# Girder assignment
Gi_Region = GirderPart.sets['Girders']
flangeg1_Region = GirderPart.sets['girder_flange_1'] #changed
flangeg2_Region = GirderPart.sets['girder_flange_2'] #changed
GirderPart.SectionAssignment(region=Gi_Region, sectionName=('Sec_girder'), offset=0.0,
    offsetType=MIDDLE_SURFACE, offsetField='', thicknessAssignment=FROM_SECTION)
GirderPart.SectionAssignment(region=flangeg1_Region,          sectionName=('Girder_flange'),
offset=0.0,
    offsetType=MIDDLE_SURFACE,  offsetField='',  thicknessAssignment=FROM_SECTION)
#newchange
GirderPart.SectionAssignment(region=flangeg2_Region,          sectionName=('Girder_flange'),
offset=0.0,
    offsetType=MIDDLE_SURFACE,  offsetField='',  thicknessAssignment=FROM_SECTION)
#newchange
```

```
==========================================================================

# Create instances for Girder and panel

==========================================================================

import assembly

# strake

Panel_Assembly = StrakeModel.rootAssembly

Panel_Assembly.DatumCsysByDefault(CARTESIAN)

#Changed: Panel and girder assembly

Str_Instance = Panel_Assembly.Instance(name='Panel', part=PanelPart, dependent=OFF)

Girder_Instance = Panel_Assembly.Instance(name='Girder', part=GirderPart, dependent=OFF)

Panel_Assembly.rotate(instanceList=('Girder', ), axisPoint=(0.0, 0.0, 0.0), axisDirection=(0.0, 1.0,
0.0), angle=90.0)

Panel_Assembly.translate(instanceList=('Girder', ), vector=(0.0, 0.0, (depth*(depthmp-1))))

# Merge panel and girders

Panel_Assembly.InstanceFromBooleanMerge(name='Grillage',
instances=(Panel_Assembly.instances['Girder'],
   Panel_Assembly.instances['Panel'], ), originalInstances=DELETE, domain=GEOMETRY)

==========================================================================

# # # Create edge sets for boundary conditions

==========================================================================

GrillageInstance=Panel_Assembly.instances['Grillage-1']

PartEd = GrillageInstance.edges

# Edge along z axis

leftE = PartEd.getByBoundingBox(-0.1, 0., -0.1, 0.01, webheight+0.01, (depth*(depthmp))+0.1)
#MKmod: depthmp-1 modified to depthmp

Panel_Assembly.Set(edges=leftE, name='Leftedge')

rightE = PartEd.getByBoundingBox(sum(st_w)-0.01, 0., -0.1, sum(st_w)+0.01, webheight+0.01,
(depth*(depthmp))+0.1)

Panel_Assembly.Set(edges=rightE, name='Rightedge')

# # Edge along x axis (transverse to frames)

bottomE = PartEd.getByBoundingBox(-0.1, 0., -0.1, sum(st_w)+0.01, webheight+0.01, 0.01)

Panel_Assembly.Set(edges=bottomE, name='bottomedge')

topEdge = PartEd.getByBoundingBox(-0.1, -0.01, (depth*(depthmp))-0.01, sum(st_w)+0.01,
webheight+0.01, (depth*(depthmp))+0.01)
```

```
Panel_Assembly.Set(edges=topEdge, name='topEdge')
# ==========================================================================
#                              Meshrefinement box
# ==========================================================================
import mesh
import regionToolset
Panel_Assembly.makeIndependent(instances=(Panel_Assembly.instances['Grillage-1'], ))
session.viewports['Viewport: 1'].view.setValues(session.views['Top'])
session.viewports['Viewport: 1'].view.setProjection(projection=PARALLEL)
f1 = GrillageInstance.faces
e1 = GrillageInstance.edges
t  =  Panel_Assembly.MakeSketchTransform(sketchPlane=f1.findAt(coordinates=(sum(st_w)/2.,
0.0,
   3.333333), normal=(0.0, -1.0, 0.0)), sketchUpEdge=e1.findAt(coordinates=(
   sum(st_w), 0.0, depthmp*depth/2.)), sketchPlaneSide=SIDE1, origin=(0, 0.0, 0))
s1 = mdb.models['Model-1'].ConstrainedSketch(name='__profile__',
   sheetSize=sum(st_w), gridSpacing=spacing[0]/2., transform=t)
g, v, d, c = s1.geometry, s1.vertices, s1.dimensions, s1.constraints
s1.setPrimaryObject(option=SUPERIMPOSE)
session.viewports['Viewport: 1'].view.fitView()
# ------------------------------------------------Test---------------------------------------
# GrillageInstance=Panel_Assembly.instances['Grillage-1'] defined already
pickedFaces  = GrillageInstance.faces
e21=GrillageInstance.edges
# Coordinates of the refined patch
# s1.rectangle(point1=(8.935, 4.65), point2=(4.375,7.95))
s1.rectangle(point1=(pxz[0], pxz[2]), point2=(pxz[1], pxz[3]))
Panel_Assembly.PartitionFaceBySketchThruAll(sketchPlane=pickedFaces.findAt(coordinates=(
sum(st_w)/2.,
   0.0, 3.333333), normal=(0.0, -1.0, 0.0)), sketchUpEdge=e21.findAt(
   coordinates=(sum(st_w),  0.0,  depthmp*depth/2.)),  faces=pickedFaces,
sketchPlaneSide=SIDE1,
   sketch=s1)
```

```
s1.unsetPrimaryObject()
if pathclocation=='plate' or pathclocation=='frame':
    execfile('Patch-350x350.py') #newchange
============================================================================

# Create mesh
============================================================================

# use the existing sets and define faces based on those to set mesh controls for all elements
# except girders
x=[]
x.append(GrillageInstance.sets['Webs'].faces)
x.append(GrillageInstance.sets['web_flange_1'].faces) #changed
x.append(GrillageInstance.sets['web_flange_2'].faces) #changed
for i in range(n_str):
    x.append(GrillageInstance.sets[('Flange1_set'+ str(i+1))].faces) #changed
    x.append(GrillageInstance.sets[('Flange2_set'+ str(i+1))].faces) #changed
    x.append(GrillageInstance.sets[('frame_set'+ str(i+1))].faces)
    x.append(GrillageInstance.sets[('shell_set'+ str(i+1))].faces)
# for i in range(len(x)):
for i in range(0, 1): #changed
    Panel_Assembly.setMeshControls(regions=x[i], elemShape=QUAD)
for i in range(1, len(x)): #changed
    Panel_Assembly.setMeshControls(regions=x[i], elemShape=QUAD,
technique=FREE)#STRUCTURED)
# Assign element type to whole instance
elemType = mesh.ElemType(elemCode=S4R, elemLibrary=EXPLICIT,
    secondOrderAccuracy=OFF, hourglassControl=DEFAULT, elemDeletion=ON,
maxDegradation=1.0)


Gri_faces=GrillageInstance.faces
GrillageInstance_Faces_Region = (Gri_faces,)
Panel_Assembly.setElementType(regions=GrillageInstance_Faces_Region,
elemTypes=(elemType,))
# # Seed whole instance edges with one mesh size
```

```
Grillage_Instance_Edges = GrillageInstance.edges
Panel_Assembly.seedEdgeBySize(edges=Grillage_Instance_Edges, size=mesh_size)
Refined=Grillage_Instance_Edges.getByBoundingBox(pxz[0]-tol,-tol,pxz[2]-
tol,pxz[1]+tol,webheight+tol,pxz[3]+tol)
Panel_Assembly.seedEdgeBySize(edges=Refined, size=mesh_fine)
Refined_face=Gri_faces.getByBoundingBox(pxz[0]-tol,-tol,pxz[2]-
tol,pxz[1]+tol,webheight+tol,pxz[3]+tol)
Refined_Region = (Refined_face,)
Panel_Assembly.generateMesh(regions=Refined_Region)
  # Create a set for rest of the structure (excluding refined)
# START SET CREATION ----------------------------------------------
# Create a set for refined region
refset=Panel_Assembly.Set(faces=Refined_face,name=('refined'))
Full=Gri_faces.getByBoundingBox(-1,-1,-1,25,25,25)
fullset=Panel_Assembly.Set(faces=Full,name=('full'))
# Create a set by substracting one set from another
Panel_Assembly.SetByBoolean(name=('minus'),sets=(fullset,refset),operation=DIFFERENCE)
# Create a minus region
minusregion=(Panel_Assembly.sets['minus'].faces,)
pickedRegions = Gri_faces.findAt(((pxz[0]-tol, 0.0, depth1/2.), ), ((pxz[1]+tol,0.0, depth1/2.), ))
Panel_Assembly.setMeshControls(regions=pickedRegions, technique=FREE,
allowMapped=False)
Panel_Assembly.generateMesh(regions=minusregion)
# /3.10.2016/ SURFACE CREATION FOR PRESSURE APPLICATION (REFINED REGION)
-----------------
reined_set=refset.faces.getByBoundingBox(-1,-0.1,-1,25,0.002,25)
refined_out=Panel_Assembly.Set(faces=reined_set,name=('refined_out'))
Panel_Assembly.Surface(side1Faces=reined_set,name='refined_out_surf')
# /3.10.2016/ SET CREATION for damage output -----------------
# # Panel_Assembly.Set(faces=damageset,name=('damageset'))
# Gri_elements=GrillageInstance.elements
#          Panel_Assembly.Set(elements=(Gri_elements.getByBoundingBox(pxz[0]-1.,-0.1,pxz[2]-
1.5,pxz[1]+1.,1.5,pxz[3]+1.5)),name=('damageset2'))
# /6.10.2016/ ELEMENT SET CREATION FOR DISPLACEMENT OUTUT -----------------
```

```
# Set from another set (subset)
# xmin=pxz[0], xmax=pxz[1], zmin=pxz[2], zmax=pxz[3]
disp_set_elements=refined_out.nodes
Panel_Assembly.Set(nodes=(disp_set_elements.getByBoundingBox(pxz[0]+1.,-
0.1,pxz[2]+0.6,pxz[1]-1.,0.002,pxz[3]-0.6)),name=('DISP'))
#         disp_set_faces=refined_out.faces.getByBoundingBox(pxz[0]+0.3,-0.1,pxz[2]+0.2,pxz[1]-
0.3,0.002,pxz[3]-0.2) # xmin,ymin,zmin,xmax....
# disp_set=Panel_Assembly.Set(faces=disp_set_faces,name=('DISP'))
# END of MESH GENERATION
# Create a step
=============================================================================
import step
if implicit != 'yes':
   StrakeModel.ExplicitDynamicsStep(name='Step-1', previous='Initial',
      timePeriod=sim_time,scaleFactor=0.9)
   # Define mass scaling
   StrakeModel.steps['Step-1'].setValues(massScaling=((SEMI_AUTOMATIC,
      MODEL, AT_BEGINNING, 4.0, 0.0, None, 0, 0, 0.0, 0.0, 0, None), ))

=============================================================================
   # Create output requests
=============================================================================
   # Create field output
   StrakeModel.fieldOutputRequests['F-Output-1'].setValues(variables=(
      'S', 'MISES', 'TRIAX', 'PEEQ', 'U', 'V', 'A', 'RF', 'CSTRESS','SF', 'EVF','STATUS','P','HP'),
numIntervals=Num_Int)
   # Create history output
   StrakeModel.historyOutputRequests['H-Output-1'].setValues(
      variables=('ALLAE', 'ALLCD', 'ALLDMD', 'ALLFD', 'ALLIE', 'ALLKE',
      'ALLPD', 'ALLSE', 'ALLVD', 'ALLWK', 'ETOTAL'), numIntervals=1*Num_Int)
=============================================================================
   # Create interaction
=============================================================================
   Contact_Prop = StrakeModel.ContactProperty('IntProp-1')
```

```
    Int_Prop = StrakeModel.interactionProperties['IntProp-1']
    # Create interaction property (general contact)
    #.# Tangential behaviour
    Int_Prop.TangentialBehavior(formulation=PENALTY, directionality=ISOTROPIC,
slipRateDependency=OFF,
        pressureDependency=OFF, temperatureDependency=OFF, dependencies=0, table=((
        0.3, ), ), shearStressLimit=None, maximumElasticSlip=FRACTION,
        fraction=0.005, elasticSlipStiffness=None)
    #.# Normal behaviour
    Int_Prop.NormalBehavior(pressureOverclosure=HARD, allowSeparation=ON,
constraintEnforcementMethod=DEFAULT)
    #.# Damping
    Int_Prop.Damping(definition=DAMPING_COEFFICIENT, tangentFraction=DEFAULT,
        clearanceDependence=STEP, table=((0.01, ), ))
    # Create interaction
    Strake_Model_ContactExp = StrakeModel.ContactExp(name='Int-1', createStepName='Step-
1')
    Int_Prop2 = StrakeModel.interactions['Int-1']
    Int_Prop2.includedPairs.setValuesInStep(stepName='Step-1', useAllstar=ON)
    Int_Prop2.contactPropertyAssignments.appendInStep(
        stepName='Step-1', assignments=((GLOBAL, SELF, 'IntProp-1'), ))
elif implicit=='yes':
    mdb.models['Model-1'].StaticStep(name='Step-1', previous='Initial',
        timePeriod=2.0, maxNumInc=500, initialInc=0.04, minInc=1e-11, maxInc=0.04,
        nlgeom=ON)
=====================================================================
    # Create interaction
    Contact_Prop = StrakeModel.ContactProperty('IntProp-1')
    Int_Prop = StrakeModel.interactionProperties['IntProp-1']
    # Create interaction property (general contact)
    #.# Tangential behaviour
    Int_Prop.TangentialBehavior(formulation=PENALTY, directionality=ISOTROPIC,
slipRateDependency=OFF,
        pressureDependency=OFF, temperatureDependency=OFF, dependencies=0, table=((
```

```
    0.3, ), ), shearStressLimit=None, maximumElasticSlip=FRACTION,
    fraction=0.005, elasticSlipStiffness=None)
  #.# Normal behaviour
  Int_Prop.NormalBehavior(pressureOverclosure=HARD, allowSeparation=ON,
constraintEnforcementMethod=DEFAULT)
  #.# Damping
  Int_Prop.Damping(definition=DAMPING_COEFFICIENT, tangentFraction=DEFAULT,
    clearanceDependence=STEP, table=((0.01, ), ))
  # Create interaction
  # Strake_Model_ContactExp = StrakeModel.ContactStd(name='Int-1',
createStepName='Initial')


  # Int_Prop2 = StrakeModel.interactions['Int-1']
  # Int_Prop2.includedPairs.setValuesInStep(stepName='Initial', useAllstar=ON)
  # Int_Prop2.contactPropertyAssignments.appendInStep(
    # stepName='Initial', assignments=((GLOBAL, SELF, 'IntProp-1'), ))
  StrakeModel.fieldOutputRequests['F-Output-1'].setValues(variables=(
    'S', 'MISES', 'PE', 'PEEQ', 'U', 'RF','SF',
'STATUS','TRIAX','EVOL'),numIntervals=Num_Int)
# # If patch is defined define also loading for implicit simulation
if pathclocation=='plate' or pathclocation=='frame':
  faces=GrillageInstance.faces
  StrakeModel.TabularAmplitude(name='Amp-1', timeSpan=STEP,
    smooth=SOLVER_DEFAULT, data=((0.0, 0.0), (1.0, 1.0), (2.0, 0.0)))
  # Two ways to apply loads on surface: explicitly define the surface, which is first here. However,
since element set is already defined i use the second approach
  # ----------Approach 1-----------
  #  patch=faces.getByBoundingBox(sum(st_w)/2.-spacing[0],-0.001,depth1/2.-loadpatchheight-
tol,sum(st_w)/2.+spacing[0],0.001,depth1/2.+loadpatchheight+tol)
  # region = Panel_Assembly.Surface(side1Faces=patch, name='patchsurf')
  # ----------Approach 2-----------
  # Does not create the actual surface and this is problem in analysis
  # Create a surface region from set (set name was defined in the Patch 350x350 script)
  # regionset=(Panel_Assembly.sets['patch-wxs'].faces,)
```

```
    # StrakeModel.Pressure(name='Load-2', createStepName='Step-1',
        # region=regionset, distributionType=UNIFORM, field='', magnitude=pressure,
        # amplitude='Amp-1')
    # ----------Approach 3-----------
    # Create a surface region from set (set name was defined in the Patch 350x350 script - but this is geometry set, again doesnt work)
    # patchset=Panel_Assembly.sets['patch-wxs']; elementSets=((patchset, S2),)
    # surfpatch=Panel_Assembly.SurfaceFromElsets(name='patchsurf',elementSetSeq=elementSets)
    # region=Panel_Assembly.surfaces['patchsurf']
    # StrakeModel.Pressure(name='Load-2', createStepName='Step-1',
        # region=region, distributionType=UNIFORM, field='', magnitude=pressure,
        # amplitude='Amp-1')
    # ----------Approach 3-----------
    # this works, surface created in the PATCH350x350.py file from faces
    region=Panel_Assembly.surfaces['patchsurf']
    StrakeModel.Pressure(name='Load-1', createStepName='Step-1',
        region=region, distributionType=UNIFORM, field='', magnitude=pressure,
        amplitude='Amp-1')
if pathclocation=='wide':
    a = mdb.models['Model-1'].rootAssembly
    region = Panel_Assembly.surfaces['refined_out_surf']
    StrakeModel.Pressure(name='Load-1', createStepName='Step-1',
        region=region, distributionType=USER_DEFINED, field='', magnitude=1.0,
        amplitude=UNSET)
=========================================================================

# Create boundary conditions
# Assign boundary conditions to X-core sandwich panel [str() converts integers to strings]
r1=Panel_Assembly.sets['Leftedge']
r2=Panel_Assembly.sets['Rightedge']
r3=Panel_Assembly.sets['bottomedge']
r4=Panel_Assembly.sets['topEdge']
```

```
StrakeModel.DisplacementBC(name='BC-1', createStepName='Step-1', region=r1, u1=0.0,
u2=0.0, u3=0.0, ur1=UNSET, ur2=UNSET, ur3=UNSET,
    amplitude=UNSET, fixed=OFF, distributionType=UNIFORM, fieldName='', localCsys=None)
StrakeModel.DisplacementBC(name='BC-2', createStepName='Step-1', region=r2, u1=0.0,
u2=0.0, u3=0.0, ur1=UNSET, ur2=UNSET, ur3=UNSET,
    amplitude=UNSET, fixed=OFF, distributionType=UNIFORM, fieldName='', localCsys=None)
StrakeModel.DisplacementBC(name='BC-3', createStepName='Step-1', region=r3, u1=0.0,
u2=0.0, u3=0.0, ur1=UNSET, ur2=UNSET, ur3=UNSET,
    amplitude=UNSET, fixed=OFF, distributionType=UNIFORM, fieldName='', localCsys=None)
StrakeModel.DisplacementBC(name='BC-4', createStepName='Step-1', region=r4, u1=0.0,
u2=0.0, u3=0.0, ur1=UNSET, ur2=UNSET, ur3=UNSET,
    amplitude=UNSET, fixed=OFF, distributionType=UNIFORM, fieldName='', localCsys=None)
mdb.Job(name=ship, model='Model-1', description='', type=ANALYSIS,
    atTime=None, waitMinutes=0, waitHours=0, queue=None, memory=90,
    memoryUnits=PERCENTAGE, getMemoryFromAnalysis=True,
    explicitPrecision=SINGLE, nodalOutputPrecision=SINGLE, echoPrint=OFF,
    modelPrint=OFF, contactPrint=OFF, historyPrint=OFF, userSubroutine='',
    scratch='', multiprocessingMode=DEFAULT, numCpus=1, numGPUs=0)
mdb.jobs[ship].writeInput(consistencyChecking=OFF)
# Edit keyword to add userdefined loading
# StrakeModel.keywordBlock.synchVersions(storeNodesAndElements=False)
# StrakeModel.keywordBlock.insert(111, """
# *AMPLITUDE,NAME=RAMP
# 0.,0.,0.1,1.0,0.15,1.,0.2,0.
# ** LOADS
# **
# ** Name: Load-1   Type: Pressure
# *DLOAD,AMP=RAMP
# Patch,PNU""")
```

Appendix 3: Minimum engine power requirements

Table 4.1 presents results of the calculated minimum engine power requirements for a number of sample ships, along with their input data. As *SA Agulhass II* is a new ship built in 2012, the engine power for each of its ice classes (IAS, IA and IB) has been taken to be *New ships #1, #2 and #3* (Table 4.1) respectively. For *Uikku MT*, The engine power for ice classes IAS and IA have been taken to be *Existing ships #1 and #2* respectively. For calculating the engine power requirements for ice class IB of existing ships such as *Uikku MT*, equation 4.1 has been employed.

Table 4.1. Minimum engine power requirements of different ships

		Sample ship No.								
		#1	#2	#3	#4	#5	#6	#7	#8	#9
Ice class		IAS	IA	IB	IC	IAS	IAS	IA	IA	IB
α	[degree]	24	24	24	24	24	24	36	20	24
φ_1	[degree]	90	90	90	90	30	90	30	30	90
φ_2	[degree]	30	30	30	30	30	30	30	30	30
L	[m]	150	150	150	150	150	150	150	150	150
B	[m]	25	25	25	25	25	22	25	25	25
T	[m]	9	9	9	9	9	9	9	9	9
L_{BOW}	[m]	45	45	45	45	45	45	45	45	45
L_{PAR}	[m]	70	70	70	70	70	70	70	70	70
A_{wf}	[m^2]	500	500	500	500	500	500	500	500	500
D_P	[m]	5	5	5	5	5	5	5	5	5
Prop. No / type		1/CP	1/CP	1/CP	1/CP	1/CP	1/CP	1/CP	1/CP	1/FP
New ships	[kW]	7840	4941	3478	2253	6799	6406	5343	5017	3872
(see section 3.2.2)										
Existing ships	[kW]	9192	6614			8466	7645	6614	6614	
(calculated according										
to the formula in 3.2.4)										

The engine output for an existing ship of ice class IB shall not be less than that determined by:

$$P = f_1 . f_2 . f_3 . (f_4 \Delta + P_0) \text{ [kW]} \quad \text{..... (4.1)}$$

where

f_1 = 1.0 for a fixed pitch propeller

= 0.9 for a controllable pitch propeller

$f_2 = \Phi_1 / 200 + 0.675$ but not more than 1.1,

where

Φ_1 is the rake of the stem at the centerline [degrees]

The product $f_1 f_2$ shall not be taken as less than 0.85.

f_2 = 1.1 for a bulbous bow

$f_3 = 1.2B/\ \Delta^{1/3}$ but not less than 1.0

f_4 and P_o shall be taken as follows:

Ice class	IB	IC	IB	IC
Displacement	Δ < 30 000		Δ ≥ 30 000	
f_4	0.22	0.18	0.13	0.11
P_0	370	0	3070	2100

Δ is displacement [t] of the ship on the maximum ice class draught. It need not be taken as greater than 80 000 t.