Assessment of structural requirements related to LNG fuel tanks

Halvor Larsson Aga

Marine Technology
Submission date: June 2013
Supervisor: Bernt Johan Leira, IMT
Co-supervisor: Hans Arne Blomvågnes-Bakke, Rolls-Royce Marine AS
Sören Ehlers, IMT

Norwegian University of Science and Technology
Department of Marine Technology

Master Thesis, Spring 2013
for
Stud. Techn. Halvor L. Aga

Assessment of structural requirements related to LNG fuel tanks

Vurdering av konstruksjonskrav relatert til drivstoff-tanker for LNG

The maritime industry is always on the lookout for more profitable and environmentally friendly shipping solutions. One of the latest trends is the usage of liquefied natural gas, LNG, for propulsion. The transformation from oil to gas powering requires close attention to new safety hazards and introduction of new barriers and safety measures. One of the key differences is the storage of the liquefied gas, which cannot be contained within the traditional tank holds. Special structurally independent tanks are used, and these need to be placed in a secure location on the ship, so that any consequences from an incident could be minimized, or at best totally avoided.

The objective of this thesis is to assess the safety of the surrounding structure with regards to another ship colliding into the side of the ship at the location of the enclosed LNG fuel tanks. New classification rules may give room for closer placement to the ships side than the present ones. A requirement for doing this will most probably be an analysis regarding the capability of the structure capability to absorb an equivalent amount of energy as compared to the standard design, when hit by another vessel. IMO are currently revising these rules and changes are expected. The new rules might give space for case-specific calculations to control the safe distances from the tank edge till the shipside.

The following subjects are to be examined:

1. Current rules for design of cargo carriers with confined LNG fuel tanks are to be reviewed. The rules "gas fuelled ship installations" from DNV shall be used as a basis. On-going rule development should also be discussed.

2. The basic model for ship collision in the tank area shall be discussed and a discussion around implications from ice class should be carried out.

3. Methods for assessment of collision energy absorption shall be reviewed. The ability to quantify this energy shall be demonstrated through the modelling of an example study. Boundary conditions and convergence should be discussed. The analysis is to be carried out using LS-DYNA.

4. The magnitude of the required energy absorption should be estimated, also considering applicable rules. Possible approaches for achieving better energy absorption should be discussed. In particular, the effect of increasing the steel thickness of the outer skin, the webframes and the stringers should be investigated. Parametric studies are to be performed to the extent that time allows based on discussion with the supervisors.

The work scope may prove to be larger than initially anticipated. Subject to approval from the supervisor, topics may be deleted from the list above or reduced in extent.

In the thesis the candidate shall present his personal contribution to the resolution of problems within the scope of the thesis work.

Theories and conclusions should be based on mathematical derivations and/or logic reasoning identifying the various steps in the deduction.

The candidate should utilise the existing possibilities for obtaining relevant literature.

The thesis should be organised in a rational manner to give a clear exposition of results, assessments, and conclusions. The text should be brief and to the point, with a clear language. Telegraphic language should be avoided.

The thesis shall contain the following elements: A text defining the scope, preface, list of contents, summary, main body of thesis, conclusions with recommendations for further work, list of symbols and acronyms, references and (optional) appendices. All figures, tables and equations shall be numbered.

The supervisor may require that the candidate, in an early stage of the work, presents a written plan for the completion of the work. The plan should include a budget for the use of computer and laboratory resources which will be charged to the department. Overruns shall be reported to the supervisor.

The original contribution of the candidate and material taken from other sources shall be clearly defined. Work from other sources shall be properly referenced using an acknowledged referencing system.

The thesis shall be submitted in electronic form:

- Signed by the candidate
- The text defining the scope included
- Drawings and/or computer prints which cannot be bound should be organised in a separate folder.

Supervisor: Professor Bernt J. Leira   Co-supervisor: Professor Sören Ehlers

Contact person in Rolls-Royce: Hans Arne Blomvågnes-Bakke

Deadline: June 10th 2013

Trondheim, January 14th, 2013

Bernt J. Leira

Preface

This is the report from a study carried out during spring semester 2013 as the finishing work for the master degree in marine structures. The subject code is TMR4900 and the work is carried out at the Norwegian University of Science and Technology. The theme is structural analysis of ship collisions.

Planning for undesired events sometimes prove to be a demanding task, many variables stay unknown because of the nature of accidents. Ship collisions represent a huge threat to a ship. The desire to contribute to the mitigation of disastrous consequences, and the fact that nonlinear numerical modelling is a tool of choice, resulted in an interest for this field of study. The practical take on the problem considered in this thesis results from the fact that it was developed in close cooperation with representatives from the industry. This has made it easy to familiarize myself with the problem at hand.

Of lessons learned, the biggest was in the parametrical modelling, which was started directly. This resulted in a lot of analyses carried out with errors, such that they needed to be re-run. A better way of carrying this work out would have been to model the section the normal way i.e. by the use of the graphical interface, and secondly made the desired variables parametrical. In this way the initial model could have been used for verification and convergence studies, and the final parametrical model could have been benchmarked against the first model. Also the number of parameters would have been drastically reduced, resulting in easier programming. All in all, this patient work has resulted in great respect for the carefulness with which the used methods must be applied and for the needed level of knowledge of the user.

I would like to thank my supervisor Prof. Bernt J. Leira, my co-supervisors Prof. Sören Ehlers and M.Sc. Hans Arne Blomvågnes-Bakke for patient supervising and discussions through the whole semester. I would also like to thank Ph.D. candidate Martin Storheim for help on simulation issues, and my fellow students for day to day discussion.
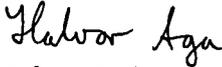
Halvor L. Aga

Trondheim 10. June 2013

Abstract

This thesis considers collision between ships, and the structural resistance of the struck ship with the aim to minimize the indentation into the ship side.

First it gives a background in ship collision analysis, different techniques are discussed, and emphasis is given to the decoupling of the ship collision problem. Secondly it assesses the rules that come into account when introducing LNG fuel tanks in a cargo ship, rules regarding modelling of ship structures as well as expected rule development. An introduction to the existing methods for estimation of energy involved in ship collisions is given. The different structural components of a ship section are discussed, and applicable analytical formulae as well as an analytical method for determining the force indentation curves for a full ship section are reviewed. Special structural elements of core type are briefly discussed.

A parametrical model of a ship side section capable of modelling different configurations for the structural layout is developed. This model is used for collision analysis by means of LS-DYNA, where displacement controlled impact with a rigid and simple cone shape indenter is driven into the side section at a right angle. The simulations are uncoupled. They are verified by means of convergence and sensitivity testing. Variables which should be given close attention enclose failure strain of the material and the static friction coefficient. The model is compared to analytical solutions, showing agreement to a certain degree. For comparative studies the model is given confidence.

The model is used for a comparative study where the aim is to assess the reduction in safe distance for a gas tank, by taking into account the effect of ice strengthening. In this case study it is concluded that the fuel tanks could be moved 1.37 metres or 1.45 metres closer to the ship side for two different implementations of ice class. Structural parameters are studied. The main conclusions are that an increase of the outer skin thickness or the introduction of an extra stringer gives good results. For concurrent increase the thicknesses in webframes, stringers and outer skin give good results. A comparison table is developed, and a comparison ratio is presented for each parametrical variation. It is noted that the ice strengthened design gives promising comparison ratios. This study has been carried out by variation over the whole height of the section and "individualization" of the variables over the height might lead to improved results.

Sammendrag (abstract in Norwegian language)

Denne avhandlingen omhandler kollisjoner mellom skip. En skipsstrukturs evne til å ta opp energi fra en kollisjon er studert med hensikten å minimere inntrengingsdybden i det trufne skroget.

Først oppsummeres bakgrunnen for analyse av skipskollisjoner, forskjellige teknikker er diskutert, med hovedvekt på ukoblede metoder. Regler som omhandler plassering av LNG forbrukstanker i lasteskip, modellering av skipsstrukturer så vel som utvikling av nye regler diskuteres. Metoder som brukes for estimering av energi i skipskollisjon diskuteres. Forskjellige strukturelementer, med formler for enkelt elementers og hele strukturers evne til å ta opp energi siteres fra litteraturen og diskuteres. Spesielle strukturelementer diskuteres kort.

En parametrisk modell for modellering av et skips halve tverrsnitt mellom to tverrskott presenteres. Denne er brukt til analyser i LS-DYNA, hvor forflytnings styrt simulering av skipskollisjon ved bruk av en kjegleformet og fast baug utføres. I disse simuleringene brukes ukoblet metode. Simuleringene verifiseres ved bruk av konvergens og sensitivitets tester. Ved å introdusere randbetingelser i nærhet til kollisjonssonen, sammenlignes kraften fra simuleringene med en analytisk modell. Kreftene viser likhet i den grad det kan forventes. Modellen gis tillit for sammenligningsstudier.

Videre utføres sammenligningsstudier hvor målet er å undersøke reduksjonen i nødvendig distanse fra gass tank til skip side i tilfeller hvor is forsterking av skroget er utført. Et eksempel studie presenteres, og konklusjonen fra denne er at gasstanken for den gitte studien kan flyttes ut 1.37m i tilfellet hvor vertikale is rammer er implementert og 1.45 meter i tilfellet hvor langsgående stivere er implementert. Videre studeres strukturelle parametere, og hovedkonklusjonen er at en øking av tykkelse i ytterhud, eller introduksjon av en ekstra stringer så vel som samtidig øking av tykkelse i ytterhud, stringere og webspant gir gode resultater. En sammenligningstabell utvikles, og en sammenligningsfaktor presenteres for hver parametriske variasjon. Det noteres at studiene av is forsterking gir gode sammenligningsfaktorer. Studiet idealiserer skipsstrukturen med variabler som er konstant for hele høyden, og en forbedring i sammenligningsfaktoren kan forventes dersom ulike strukturelle parametere tillates over høyden.

Abbreviations and terms

DNV	Det Norske Veritas
GL	Germanischer Lloyds
IMO	International Maritime Organisation
BLG	IMO's sub-committee on Bulk, Liquids and Gas.
IGF code	International code of safety for ships using gases and other low flashpoint fuels.
FPSO	Floating production, storage and offloading unit
LNG	Liquified Natural Gas
WF	Webframe
STR	Stringer
IS	Inner Skin
OS	Outer Skin
.ses file	Session file containing commands for PATRAN or LS-PREPOST
.key file	Keyword file including commands for LS-DYNA
NR	Newton Raphson method
RCTL	Failure criterion based on continuum formulations by Rice-Tracey and Crockcroft-Latham.
GT	Gross Tonne.
ROPAX	Combined roll on-roll off and passenger vessel.
TNO	The Dutch institute for applied physical research.
MATLAB	Program for solving mathematical problems, especially matrix formulated problems. Webpage see (1).
PATRAN	Program for pre and post processing for simulations for use with a wide range of numerical programs. Webpage see (2).
LS-DYNA	Program for solving numerical problems including nonlinear finite element method. Webpage see (3).
LS-PREPOST	Program for pre- and post-processing of analyses in LS-DYNA. Webpage see (4).

CONTENTS

1 INTRODUCTION

This section gives the motivation for the current thesis and the scope of the study.

1.1 MOTIVATION

As with cars a collision between two ships is an undesired event. In the car industry crashworthiness appears as an important quality measure. Full scale tests are carried out and different scenarios provide results which give rise to a classification by means of stars. Many engineering hours are invested to make the structure of a car able to absorb as much energy as possible, and the ranking of a car is being used as a commercial advantage for car sellers. Safety is often one of the main criteria for a customer to choose a specific car.

Traditionally ships have not been specifically engineered to reduce the consequences of a collision, and full scale testing is expensive and therefore relatively rare. The assessment of crashworthiness for a ship needs a different approach than what is used in the car industry.

Ship design is often highly specialised, making collision analysis very costly per ship as much analysis is needed for a design which might be used for a few ships only. However, in the period between 2007 and 2011 about 22% of all serious losses and 9% of all total losses for vessels over 500GT are due to collision or contact (5).

Studies of special structural elements (for example (6) and (7)) designed to absorb energy for collision as well as optimisation studies (for example (8)) and comparison studies (for example (9)) are readily available. Special structural elements are discussed in section 5 and the optimization scheme and the comparison study are discussed in section 2.

Knowledge of the basic measures that may be implemented to enhance capability to absorb energy may be useful when a crashworthy ship within the relatively conservative frames of ship design is required. Basic measures in this context are variables such as the number or thicknesses of stiffeners, stringers and web frames.

1.2 SCOPE

For a cargo ship, space is critical. The implementation of liquefied natural gas (LNG) as fuel requires safe storage inside or on the deck of the given ship. In an accident where one ship hits another, and damages the struck ship's LNG-fuel tanks, the contents might be released. This might give rise to fatal consequences making it necessary to keep the integrity of the tank as long as possible.

The class societies give scantlings for the minimum distance allowed from the outer skin to the edge of the tank. This is set in relation to the breadth of the ship ((10) and (11)). One of the measures for improving the safety against an impact of the tanks is a strengthening of the ship side. This ensures that the safety in a ship collision is kept at the same level while decreasing the distance as defined. A parametrical study might prove useful for identifying the proper structural measures to make good and viable solutions for increasing the structures resistance against colliding ships. Economic decisions regarding the relation between weight, cargo space and costs must be based on proper investigations to ensure the safety level required of the ship and its operations.

Parametrical study

A parametric study is carried out in section 7, consisting of a series of collision simulations where the absorbed energy to indentation relation of different variants of the design is presented. The case studied is a standard cargo carrier with drawings supplied by Rolls-Royce Marine. These can be seen in appendix A.

The effect of the ice strengthening on the energy absorption of a side section is investigated by a case study.

2 BACKGROUND

Many studies have been carried out since Minorsky (12) published his work on energy estimation in 1959. The increase in available computing and the natural development in the research carried out make it natural to concentrate on some of the later work. This section aims to present a brief introduction to the study of ship collision analysis.

First the works of Minorsky (12) are discussed, secondly the analysis of ship collision is discussed by addressing outer dynamics, energy sharing and coupled approaches. The third part is devoted to bow deformation, the fourth discusses an optimization scheme and the last discusses a comparison study.

2.1 MINORSKY

In the late 50's the introduction of nuclear reactors on ships made the risk potential in the case of a ship collision high. Minorsky (12) developed a model to quantify the energy in a ship-ship collision, it is stated that by photographic evidence a collision between merchant ships can be treated as inelastic. Formula 1, proposed by Minorsky, estimates the energy from a ship impact. In turn the amount of deformed steel can be estimated by using the graph in figure 1, made by Minorsky in his empirical studies of well documented ship collisions at nearly right angle. The resistance factor is obtained by adding the volumes of the deformed structural members (see (12) for exact formula).

$$Lost \quad KE = \left[\frac{M_A M_B}{1.43 M_B + 2 M_A} \right] \left(V_B \sin \theta \right)^2$$

Formula 1 - Lost kinematic energy (12)
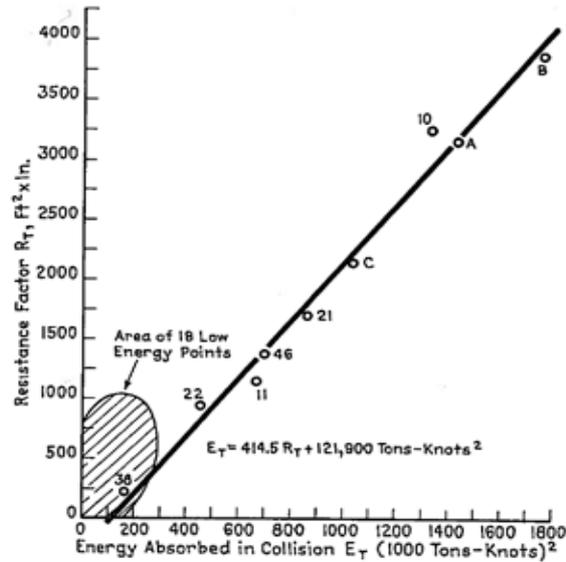
Figure 1 - Empirical energy to resistance (12)

According to Petersen (13) this method *"has been widely used since its appearance in 1959 to calculate the safety against penetration of the reactor space in nuclear powered ships and the cargo tanks in LNG-tankers."* (13). In his study formula 2 is given, describing Minorsky's empirical formula, for the damage energy in metrical units. E_D is the damage energy and R is the resistance factor describing the damaged volume.

$$E_D[MJ] = 47R[m^3] + 32[MJ]$$

Formula 2 - Minorsky's formula in metrical units (13)

To set the energies involved in a ship collision in a broader perspective we consider a ship design with a displacement of 7329 tonne and a service speed of 12.5 knots (this is the same as the ship considered in the case study in section 6 and 7). Figure 2 shows the kinetic energies released for different speeds at a right angle where two ships of equal size collide, by the use of formula 1. The struck ship is at rest or in motion, and the striking ship speed is given along the abscissa.
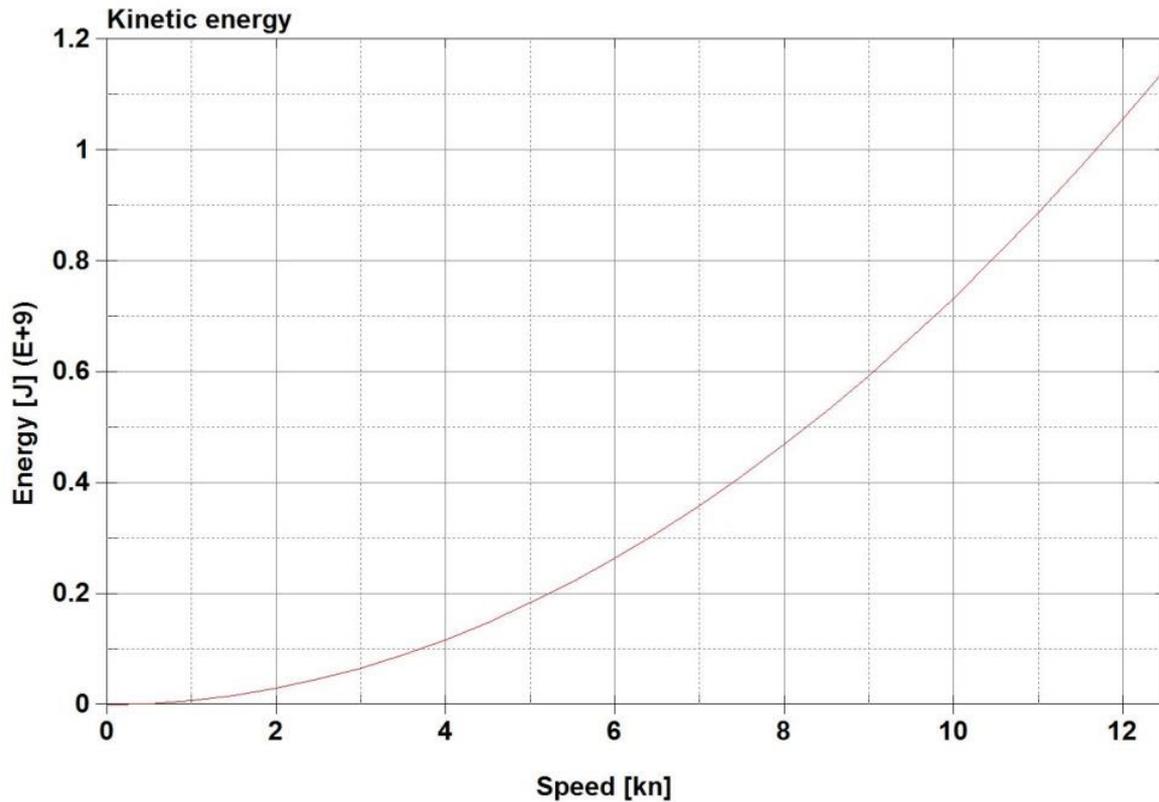
Figure 2 - Lost kinetic energy

2.2 ANALYSING THE SHIP COLLISION PROCESS

The formulae and the graph cited from Minorsky (12) illustrate a principle which is central in the analysis of ship collisions, namely the division of the analysis in two separate parts. In many cases the collision problem can be divided in two, or decoupled, Tabri (14) studies the limits for this assumption. This study is further discussed in the sub-section regarding coupled approaches.

In Minorsky's study (12) decoupling is used and the parts in this division of the problem are the outer dynamics, in the study described by formulae for estimation of collision energy, and the inner mechanics, solved by the empirical relation in figure 1. Generally the outer dynamics part analyses the ships movement and the kinetic energies by looking upon it as a totally inelastic impact, and the internal mechanics part considers the deformation, rupture and collapse of the ships involved in the collision. The solution of the inner mechanics part is discussed in the method section, and the following sub-section describes the outer dynamics.

Outer dynamics

As already mentioned the outer dynamics problem considers the movement of the colliding ships. Factors such as the added mass, point of collision, angle of collision and the speed of the inflicted vessels are central to the solution of this.

Petersen (13) describes the dynamics in an article published in 1982. This considers the horizontal movements of the involved ships. Their movement are described through the equations of motion for the ships, the use of strip method to calculate the forces from the fluid surrounding the ship, the linearization of these and a set of nonlinear springs at the point of contact.

Pedersen and Zhang (15) develop a model taking friction and sliding into account, figure 3 is cited from their study, it presents the energy ratio for two identical supply ships each at 4.5m/s forward steaming. The energy ratio is given as the energy available for crushing divided by total kinetic energy of both ships. For right angled collision it is seen that a central impact gives the greatest energy for crushing and that this is a little above 0.4 of the total kinetic energy. It is also observed that a head on collision gives more energy for deformation, and that for a right angled or near to right angled collision lower energy levels are obtained far from the centre of the struck ship.
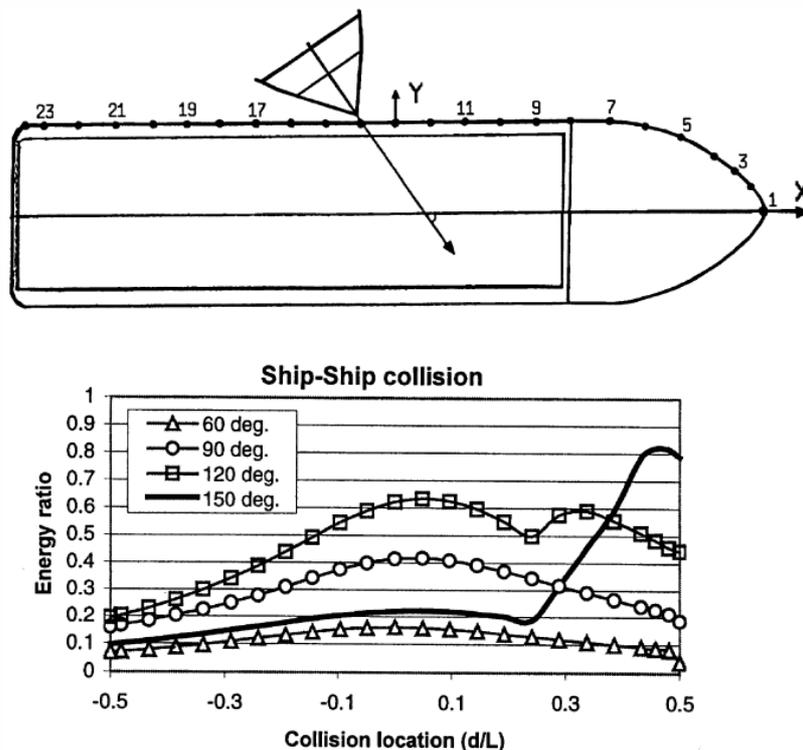
Figure 3 - Collision of two similar supply ships with equal forward speed (15)

Energy sharing

How the energy lost in a collision is shared between the involved parties depends on the structural properties of the two. For example a ship hitting a rigid wall, or as Hong et al. (16) discuss a ship hitting a platform, the ship dissipates almost the full quantity of energy. On the other hand, if a ship with a sharp and strong bow hits a bulk carrier the ship side section might deform and account for most of the energy.

The NORSOK standard (17) classifies this as ductility design, shared energy design and strength design. This is with regards to offshore installations, but it can be applicable for the understanding of the energy sharing for ships as well. Figure 4 shows the principle, this is cited from Hong et al. (16) where the study considers supply vessels striking a FPSO.
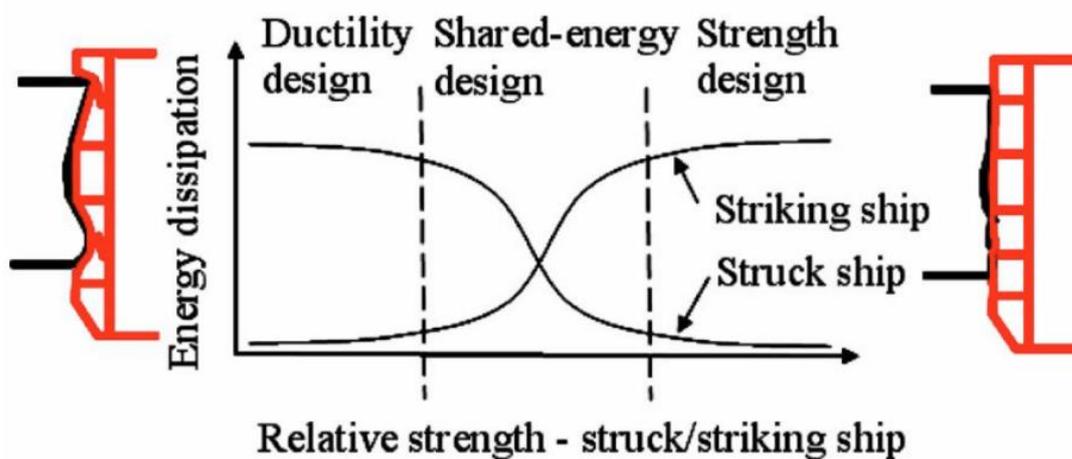
Figure 4 - Design principle (16)

Coupled approaches

The simultaneous solution of the outer dynamics and the inner mechanics problem is called a coupled approach to collision analysis. This is demonstrated by Pill and Tabri (18), where the coupling is included in a model solved by the use of LS-DYNA. The inner mechanics is solved through nonlinear finite element modelling, and the outer dynamics problem is solved by implementing mass points and a radius of gyration as shown in figure 5. The collision case they study is that of a model scale experimental study carried out by Tabri et al. (19). They show good correlation the results of this, although the prediction of the yaw motions of the struck ship were overestimated.
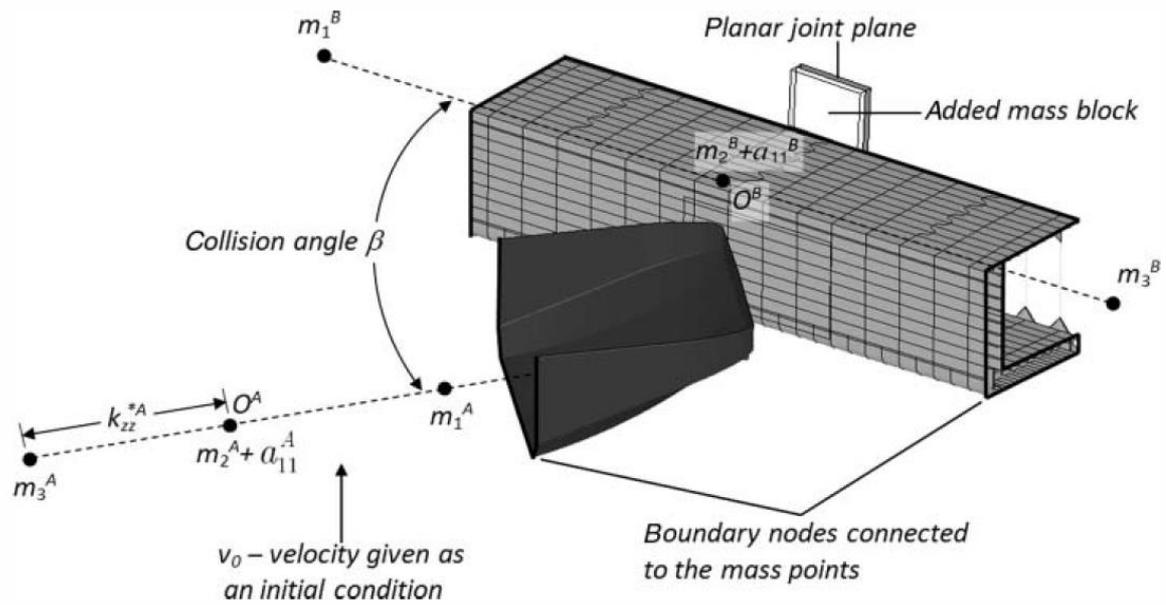
The decoupling of the collision is valid under certain assumptions, and it is convenient to use because comparison of different structures is easy to carry out. When the more precise study of a given collision case is desired a coupled approach might be needed. The boundaries for where the coupling might have a large influence are studied by Tabri (14). He concludes that in right angle collisions the decoupled approach can be used with confidence and that the precision is reduced with an increased angle from right angled. Especially the precision in penetration depth is lost at an angle.

2.3 BOW AND SIDE DEFORMATION INTERACTION

Assuming rigid behaviour of the striking bow is a common measure for quantifying a ship's ability to absorb energy. By this it is unnecessary to make assumptions about the bow structure and the numerical modelling of this is simplified. The simulations can, however, give different solutions when accounting for the bow's ability to deform. When it comes to strength design of the ship side, the bow deformation might account for huge amounts of the total energy dissipation.

In the same study as previously mentioned, Hong et al. (16), the deformations of the bow are included in their numerical model. Figure 6 shows the deformations at four time steps for a collision between bow and side section in two different cases. It illustrates that dependent of the interactions with the stem, the bulb might penetrate almost without deformation, or it might have large scale deformations.
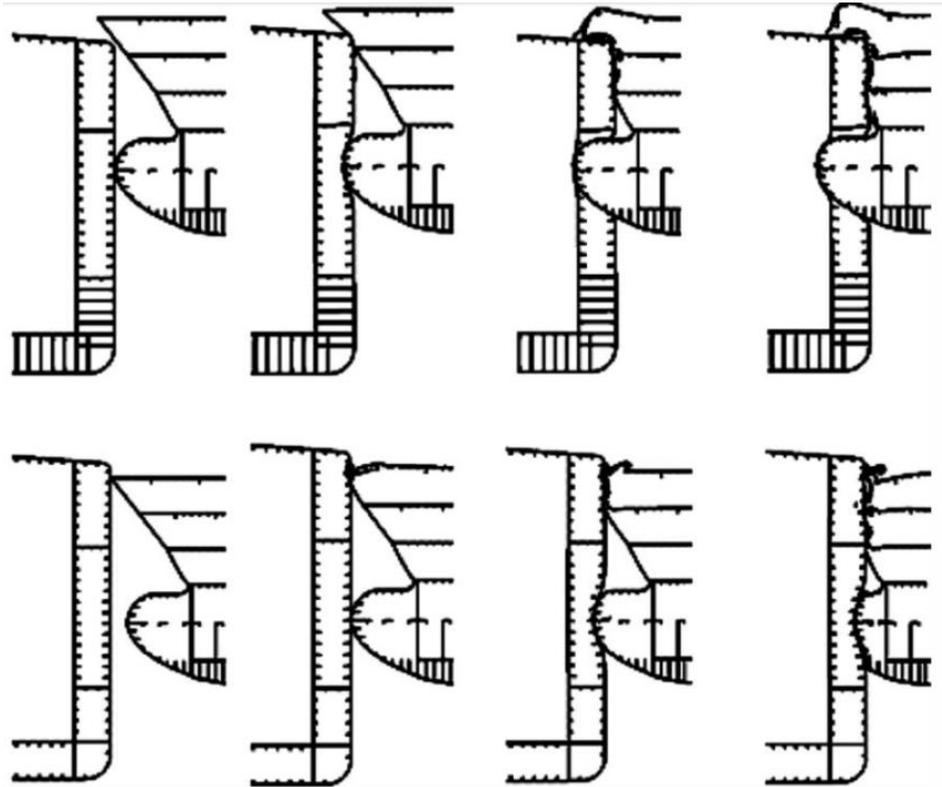
Figure 6 - Bow and side section deformation interaction (16)

2.4 OPTIMISATION PROCEDURES

Concepts of crashworthy structures have been studied by the implementation of collision analysis in a particle swarm optimization scheme by Ehlers (20). The ship side section is parameterised, i.e. giving the structural elements discrete variables, for example the thickness of a plate or the size of a stiffener. By this it is possible to define a ship section as a vector. Through the definition of a generation, as it is called in the particle swarm optimizer, a given configuration is analysed by means of LS-DYNA. This leads to a comparison value used in the optimizer for generation of the next generation. Details about particle swarm optimization can be seen in the papers by Kennedy et al. (21) and Eberhart et al. (22), and will not be discussed further in this thesis.

The scope in (20) is if the implementation of high strength steel in a LNG vessel is worthwhile when it comes to the extra cost. Figure 7 shows the evolution of the design through the generations. It is observed that it is possible to define a structure able to withstand approximately 1.8 times the collision energy at slightly increased cost and approximately the same weight, and at even lower cost of repairing collision damage.
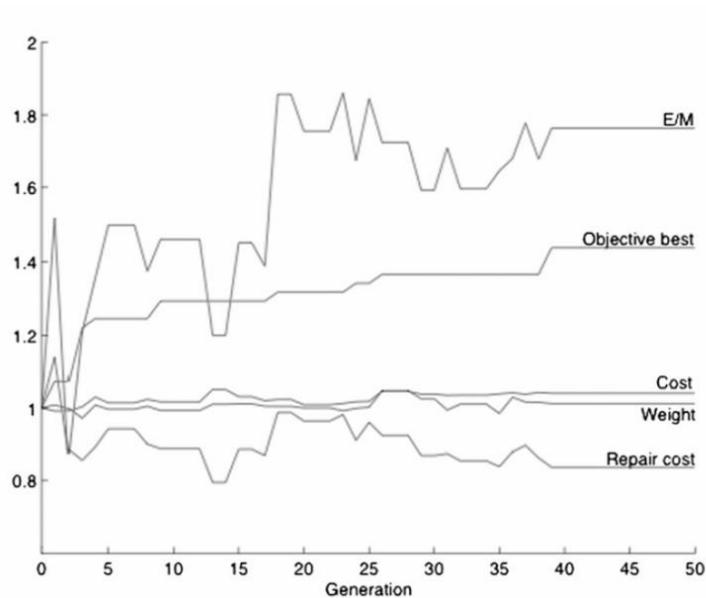
Figure 7 - Development in the optimisation scheme (20)

In (20) it is concluded that high strength steel can be a good measure for increased crashworthiness. It is noted that the optimization scheme ran over 31 days. This study also illustrates the potential increase in crashworthiness by redistribution of weight in the ship section.

2.5 COMPARATIVE STUDY

Tanker safety in the collision case has been studied by Kitamura (9). He studies the introduction of different elements in the side section for increased crashworthiness. Studied elements include increased steel quality, additional stringers, introducing a top side tank, introducing a strut in the cargo hold, a unidirectional stiffening system and, by introduction of a frame panel, a new design of the double side. They conclude that the effect of the design alterations for the unidirectional stiffening system is 7% and for the new double side design it is 10%. Effect is defined as the extra energy absorption for the change in design as opposed to assuming a proportional correlation between added steel and increased energy absorption.

3 RULE ASSESSMENT

This section is dedicated to the rules that come into account for enclosed LNG fuel tanks and modelling of ship structures. In this thesis the rules reviewed are those from DNV (Det Norske Veritas), which can all be found through the web portal (23). Some rules from GL (Germanischer Lloyds) are also included in this review.

First the rules regarding placement of LNG fuel tanks are assessed, secondly rules for collision assessment are discussed and lastly on-going rule development is described.

3.1 PLACEMENT OF LNG FUEL TANKS

DNV regulates the position of the LNG fuel tanks in enclose spaces in the rules ((10) section 3 H502), where minimum scantlings are given. Current rules regarding the horizontal placement of LNG fuel tanks state that the tanks position shall be as near to the centreline as possible and at least the minimum of:

$$\min \begin{cases} B/5 \\ 11.5m \end{cases}$$

Formula 3 - Minimum inboard distances (10)

B is the breadth of the ship, and the distance is to be measured at a right angle, at the level of the summer water line and inboard from the outer shells moulded line. This measure is subject to a possibility for reduction, by acceptance and approval from the class society. It is noted that this possibility does not apply for tanks in passenger ships or below a cargo ship's accommodation.

Dependent on the size of the tanks applied the following minimum distances also comes into account:

$$d = \begin{cases} 0.80m & for & Vc \leq 1000m^3 \\ 0.75m + Vc \cdot \dfrac{0.20}{4000}\dfrac{1}{m^2} & for & 1000m^3 \leq Vc \leq 5000m^3 \\ 0.8m + Vc \cdot \dfrac{Vc}{25000}m & for & 5000m^3 \leq Vc \leq 30000m^3 \\ 2m & for & 30000m^3 \leq Vc \end{cases}$$

Formula 4 - Lowest minimum inbord distance (10)

where Vc is the designed full gross volume of each gas tank at 20℃.

GL have similar, but not equal regulations ((11) chapter 2.8.4.2.).

3.2 RULES FOR ASSESSMENT OF COLLISIONS

Collision class

Classification societies have different ways of assessing collision risk for ships and offshore structures. Rules for quantification of collision energy for ships have not been found from DNV. GL, however, give detailed rules and the possibility for ship owners to achieve a collision class notation ((24), section 33). According to these rules two different bows are to be considered, in addition to four different drafts for each. With one analysis consisting of minimum two designs, reference and strengthened design, this makes for at least sixteen analyses. The tools acceptable include both analytical and numerical, although analytical tools are restricted to Minorsky's method (12) in high energy collisions *"if the bow and side structures are found suitable"* (24). Some of the demands from the GL rules are: the striking ship is to be of approximately the same size as the struck, the struck ship is floating freely without speed and that the bow structure is not *"extremely fully shaped"* (24).

Modelling of cut-outs

In a ship section access to the different tanks or compartments is needed, which means that cut-out or manholes are present. Following an article published by Zhang et al. for GL (25), regarding collision analysis with respect to approval of alternative arrangements, *"Cut outs and manholes in collision areas shall be taken into account during the idealization"* (25). In a submission to the BLG (IMO's subcommittee on Bulk, Liquids and Gasses) by GL, it is specified that; *"cut-outs and manholes in collision areas shall be modelled"* (26) with regard to collision analysis by means of finite element method. Premature rupture might be a problem when cut-outs are geometrically included, due to small elements around them. An alternative is to model cut-outs by the reduction of the plate thicknesses as is common in linear analysis. Formula 5 shows this process for girders in the cargo tank analysis as found in DNV class note 31.3 (27). It is stated that cut-outs affecting the *"overall force distribution or stiffness of the girder"* (27) must be modelled either geometrically or by reduction of plate thickness.

$$t_{mean} = \frac{h - h_{co}}{h \cdot r_{co}} \cdot t_w$$

$$r_{co} = 1 + \frac{l_{co}^2}{2.6(h - h_{co})^2}$$

Formula 5 - Reduction in plate thickness (27)

where t_w is the thickness of the girder web, h is the height of the girder web, h_{co} is the height of the cut-out and l_{co} is the length of the cut-out.

Limits for the use is advised as follows: with r_{co} larger than 1.2, the cut-out should be accounted for, and with r_{co} larger than 2 it should be geometrically modelled. (27)

Magnitudes of collision energy
For dimensioning it would be practical if one or more generic collision cases were defined, such that these could be considered in the simulations, for example a set magnitude of collision energy to be absorbed by a ship of a given size.

Although standards for the magnitude of accidental collision loads regarding offshore installations is given by DNV, the same for ships has not been found. Given in a DNV standard ((28) section 2 D300) is both a set minimum value for the energy to be considered and a formula for estimating the kinetic energy in other cases. Currently the minimum kinetic energy to be considered in collision analysis is 14MJ for sideways impact and 11MJ for stern or bow impact. This is equivalent to a ship with 5000 tonne displacement colliding in 2 m/s. Similarly GL regulates the same measure in (29).

3.3 DEVELOPMENT OF THE IGF CODE
Gas engines are relatively new in maritime transportation, and it is expected that eventually more ships will have this. The rules regarding the gas arrangements are not fully developed and there are expected rule changes in the following years. Through Lloyds Register's web portal (30) this progress can be assessed.

IGF code
Abbreviated the IGF Code, the "International code of safety for ships using gases and other low flashpoint fuels" are under development and have an expected entry into force in January 2016 (30). These will regulate the positioning of the LNG fuel tanks. Janse (26) propose an assessment procedure for this. Upon the 17th meeting of the IMO (International Maritime Organisation) sub-committee on bulk, liquids and gas,

abbreviated BLG and with the responsibility to develop the IGF code, (26) is submitted by GL.

First in (26) a probabilistic approach is suggested. By the reduction of the tank size, thus an increased number of tanks and clever positioning of these the consequence is reduced. Thereby the risk is kept at a constant level while reducing the distance from the outer skin to the tank. The second includes strengthening of the ship side as a barrier, and by an assessment procedure it is made sure that the same, or higher, level of energy can be absorbed before the striking ship would hit the tank in the new position. In the assessment procedure the methodology is comparable to that found in the rules for collision class by GL (24). In the proposal it seems to be the intention to allow for combinations of the two barriers; smaller tanks and stronger ship side.

4 METHOD

Analysing a ship collision can be approached in several ways: experimentally, analytically, empirically, numerically or in a combination of these. The different approaches vary both in precision of the obtained solution and the effort demanded to get this.

This section first examines the numerical approach, secondly a brief discussion of analytical methods and lastly empirical and experimental methods are discussed.

4.1 NUMERICAL METHODS

Computational resources are becoming cheaper and cheaper, making numerical modelling of complex nonlinear structural problems a viable option.

In contradiction to analytical methods, numerical methods do not provide mathematical formulae with symbols for direct calculation by insertion. The finite element method is an example of a numerical method, where a problem is divided in pieces and solved by matrix algebra. The solution from a numerical model is always an approximation of the true solution. However the error might be small and conservative, making the solution from a numerical model applicable. Ability to identify errors and erroneous solutions is of the essence, thus the level of knowledge of the users of such a model is critical for safe applications of the solutions obtained. For more information on the finite element method it is referred to (31), (32) and (33).

Expected in this thesis are many types of nonlinearities. Different types of nonlinearities are defined, Wriggers (31) list the following:

- Geometrical nonlinearities.
- Large deformations.
- Physical nonlinearities.
- Stability problems.
- Nonlinear boundary conditions.
- Coupled problems.

When solving a nonlinear problem, the identification of the expected nonlinearities is needed, and the tools needed to take each of these into account must be properly implemented.

As stated, numerical models assume that a problem can be divided into pieces, and that each of these pieces is described by mathematical relations, and that they can be connected to describe the physical problem at hand as a whole. The pieces used in finite element analysis are elements of a given size and geometry.

Shell elements

Ship sections normally have large plate fields which are described by shell elements. Through the studies of different ship collision articles, the shell formulation named "Belytscho-Lin-Tsay" seems to be the element of choice for such studies (it is used in for example (8) and (34)). Information about this element is found in the LS-DYNA theory manual (35). The understanding of the elements used is of importance in the modelling work, thus two articles which have contributed to the derivation of this element have been reviewed. These are *"Reduced and selective integration techniques in the finite element analysis of plates"* by Hughes et al. (36) and *"explicit algorithms for the nonlinear dynamics of shells"* by Belytscho et al. (37). For the derivation of basic shell element formulations for use in linear analysis it is referred to Moan ((32) section 7.4.).

Hughes et al. proposed the use of a bilinear shell element with one by one uniform reduced integration, called U1 in (36) where different elements are compared. This element only has 4 nodes, with three degrees of freedom; w, ϑ_1 and ϑ_2. These are the out of plane translation and in-plane bending terms, respectively. Totally this equals twelve degrees of freedom for one element, which is a simple plate element. This element builds on the Mindlin plate theory for thick plates, by this it follows that C^0 continuity is sufficient for the description of displacements and rotations.
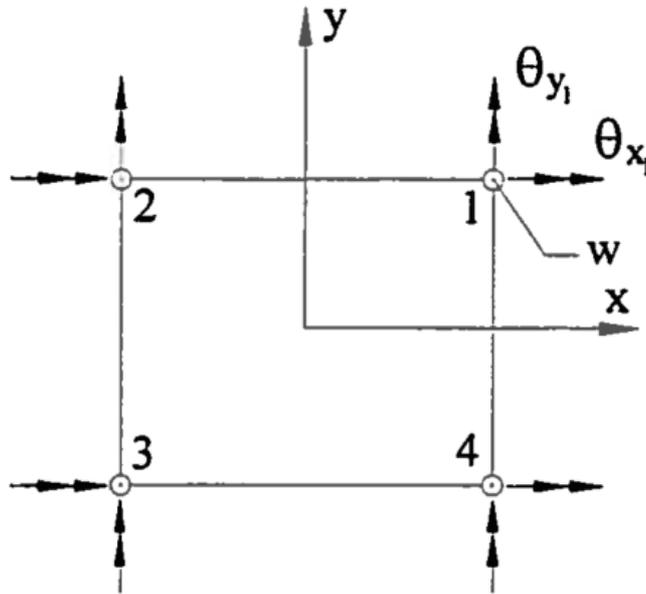
Figure 8 - Quadrilateral element with 4 nodes (32)

An interesting feature of element as presented in (36) is the use of one by one uniform reduced integration. This means that the only point of integration is at the centre of the element. It is used to avoid shear locking and to reduce the number of calculations needed to obtain a solution, but demands care and attention in its usage. For the rigid body motion of an element there is no change in strain energy. It is, however, necessary to show carefulness when it comes to what is named "spurious zero-energy modes". These are states in which the strain energy of a deformed element evaluated by the use of reduced integration will be zero, when it really is not.

Formula 6 is the strain energy formulation, where U is the strain energy, V is volume of the evaluated element and W is the strain energy per unit volume:

$$U = \int_V W dV$$

Formula 6 - Strain energy (33)

When evaluation of the strain energy is done at centre only, the strain energy and the strain per unit at centre are proportional. Neatly illustrated by figure 9 is the zero energy mode connected to the out of plane translation, w. For the discussed element there are three more, namely for both the drilling degrees as well as a twisting mode, see (36) for closer description.
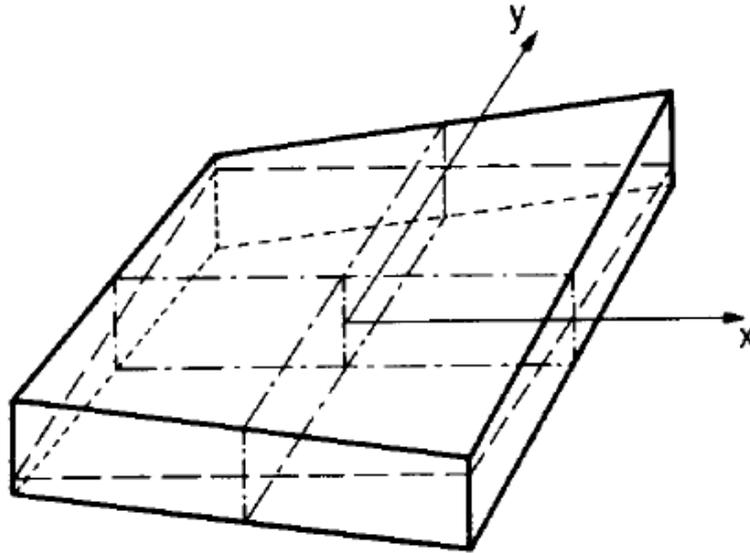
To take care of the problem regarding spurious zero energy modes, hourglass control is implemented. Belytschko et al. (37) demonstrates and validates the use of the element proposed by Hughes et al. (36) combined with a hourglass control presented by Flanagan et al. (38). In (37) also a co-rotational coordinate system is applied, meaning that the elements have a local coordinate system moving with the elements.

According to the LS-DYNA theory manual (35), the fact that the Belytscko-Lin-Tsay element has a flat geometry, leads to a weakness of its inaccuracy when describing warped configurations. This might be of significance, and an improved formulation is available for use in cases where this problem appears. Namely the *"Belytschko-Wong-Chong"* (35) improved element which includes the improvements needed for a more accurate description of warped configurations.

Beam elements

For the modelling of stiffeners, beam elements are used. The Hughes-Liu beam element is implemented in LS-DYNA (35). It is geometrically a degeneration of a solid element, and it is based on the Hughes-Liu shell, proposed by Hughes and Liu in (39) and (40).

According to LS-DYNA Theory manual (35) the following desirable qualities are included in the formulations of the element: incremental objectivity, meaning that no strains are obtained from rigid motions, simplicity, compatibility with brick elements and inclusion of transverse shear strains.

4.1.2 SOLUTION METHODS

When solving a nonlinear structural problem the response might be dependent on different nonlinear variables, such as material behaviour and changing boundary conditions.

Nonlinear static problems can often be solved by the use of iterative and incremental techniques. An example of an iterative technique is the Newton-Raphson method. This iterates on the difference between the applied force and the structural response until near to equilibrium is ensured. It updates the stiffness of the system during every iteration. The load level might be incremented by for example the Euler-Cauchy method, which increase the load in steps. It does not on itself ensure equilibrium. Combinations of iterative and incremental techniques are commonly referred to as combined methods. It is referred to (32) for more information on these techniques.

Time integration

(31) and (32) has been conferred in this section.

An alternative method for solving nonlinear systems involves the solution of the equations of motion, as formula 7 show.

$$M\ddot{r}(t) + C\dot{r}(t) + Kr(t) = R(t)$$

Formula 7 - Dynamic equation of motion (32)

Different solution methods apply for time integration, and they can be divided in two namely; explicit and implicit time integration.

1. Explicit time integration makes an assumption for how the position, velocity and acceleration will change during a small time-step. With this inserted into the equation of motion the values of the next time step can be found. All that is needed is the named values for the previous time-step. This method needs small time-steps for stability and is thus best suited for analyses of short duration. The central difference method is discussed in chapter 4.1.4 Solver.

2. Implicit time integration makes an assumption regarding the velocities and accelerations at a future time to obtain the displacements. This method is stable for longer steps and is well suited for long time analysis. This method can be formulated to give unconditionally stable solutions.

4.1.3 MATERIAL MODELS

The nonlinear behaviour of materials often plays an essential part in nonlinear modelling. The material model chosen may lead to huge differences in the results. The first part of the nonlinear material model is the stress strain relation.

Stress strain relations

A simple approach is to assume an elastic-perfectly plastic material model. This model leads to conservative results, as it does not assume any hardening of the material. Figure 10 shows an example of such behaviour.
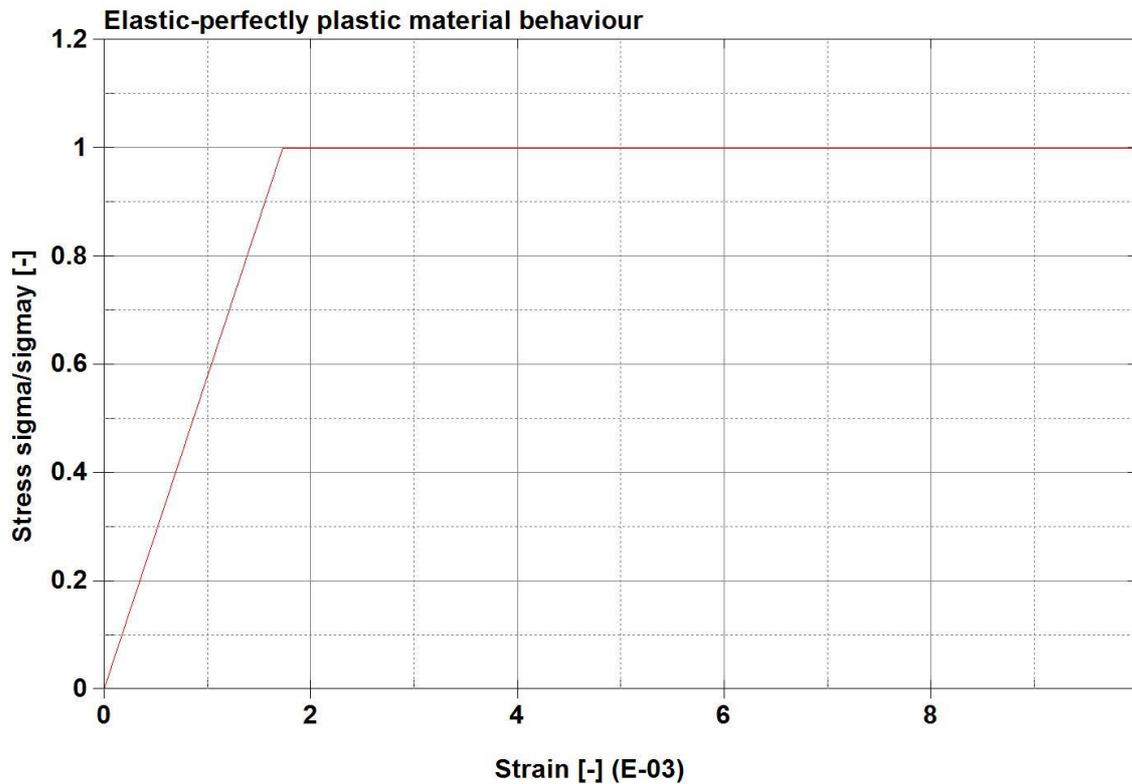
Figure 10 - Elastic-perfectly plastic material behaviour

A more demanding but refined way to describe this is shown by formula 8, cited from Alsos et al. (41).

$$\sigma_{eq} = \begin{cases} \sigma_Y & if & \varepsilon_{eq} \leq \varepsilon_{plat} \\ K(\varepsilon_{eq} + \varepsilon_0)^n & otherwise \end{cases}$$

$$\varepsilon_0 = \left(\frac{\sigma_Y}{K}\right)^{\frac{1}{n}} - \varepsilon_{plat}$$

Formula 8 – Stress strain formulas (41)

By the use of the values in table 1 and formula 8, the stress strain relation as shown in figure 11 is obtained:

Table 1 - Material parameters (41)

σ_Y	285Mpa
K	740Mpa
n	0.24
ε_{plat}	-
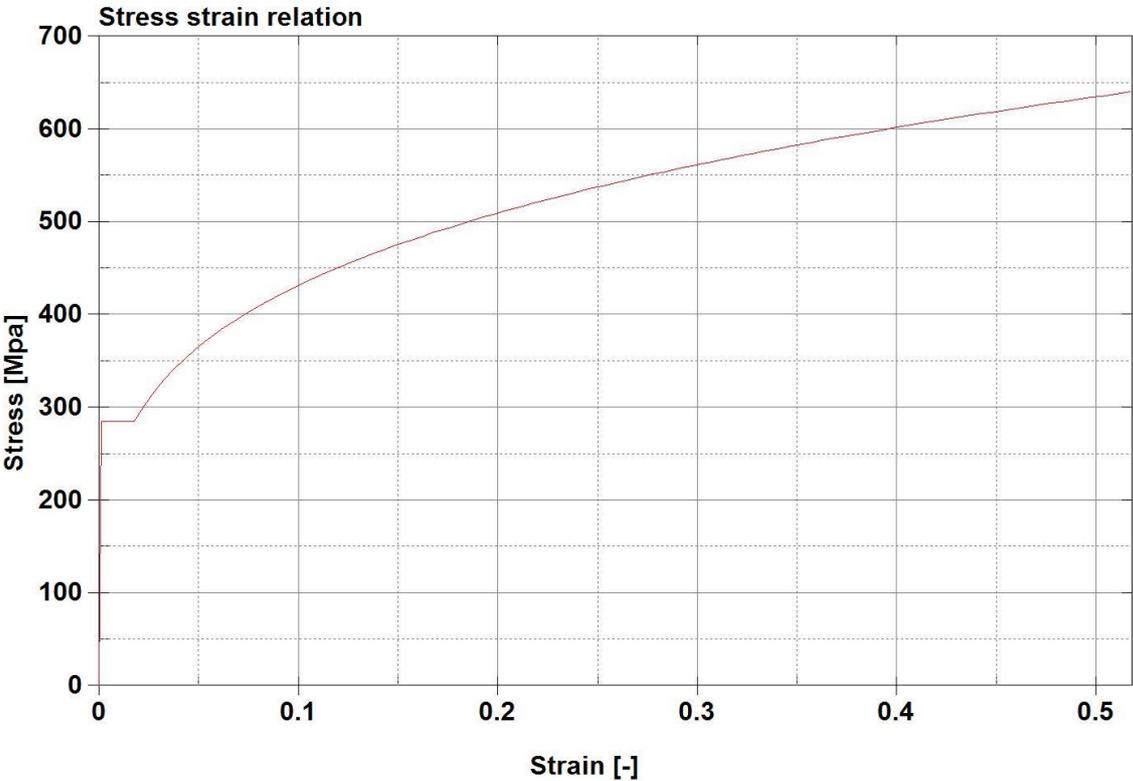
Figure 11 - Stress strain relation

Failure criteria

The other part of the material definition is the strain needed for onset of fracture. This is dependent on the smallest element length. In figure 12, a critical strain to element length curve is cited as presented by Ehlers (8):

In a study by Ehlers et al. (42) three different failure criteria are compared in a benchmark test. The failure criteria are the through thickness strain criterion, a criterion proposed by Peschmann and the RCTL criterion.

Through thickness strain means thinning strain, an element is erased if its through thickness strain reaches a failure limit. The following empirical criterion for this limit is proposed by GL:

$$\varepsilon_f(l_e) = \varepsilon_g + \varepsilon_e \cdot \frac{t}{l_e}$$

where t denotes thickness, l_e the element length, ε_g uniform strain and ε_e the necking strain.

The Peschmann criterion is experimentally obtained, while RTCL criterion originates from two continuum damage models. It is referred to Ehlers et al. (42) for further details.

Welds

Although Ohtsubo et al. (43) consider welding in their study, weld modelling is commonly not considered in nonlinear finite element analysis of ship structures. This can be seen for example in studies from Ehlers et al. (42) and Zheng et al. (44).

4.1.4 SOLVER

LS-DYNA is an example of a solver used in collision analysis. It has been validated by for example Wu et al. (45) in a benchmark study and is commonly used (for example by Pill et al. (18) and Haris and Amdahl (34).).

It includes both implicit and explicit solvers, but only the explicit will be discussed here. The input file is a "keyword" file, meaning that it is structured by keywords or commands and following values for the given keyword. For reference the LS-DYNA keyword manual (46) can be conferred. All theoretical basis for the code can be viewed in the LS-DYNA theory manual (35).

Generally all that is discussed in section 4.1 is easily implemented in the keyword file. This section will further discuss the contact algorithm and time integration scheme used by LS-DYNA.

Contact algorithm

The LS-DYNA Theory manual (35) state that automatic definition of contact is possible, and by defining the indenter as slave and all parts of the ship section as master the contact is taken care of by built in algorithms. Implemented in LS-DYNA are three different ways of treating contact and impact. From these the penalty method will be discussed in brief. This method use forces, which can be seen as springs, between all nodes and surfaces that are penetrated. Forces are applied normal to the surface and are by the *"standard penalty formulation"* determined by proportionality to the distance of penetration. To determine where such forces are applied, "*slave search*" is applied. This is used in all contact algorithms, and finds the closest point on the master surface for each slave node.

Time integration

LS-DYNA employ explicit central difference for integration of the equations of motion (35). This scheme solves the semi discrete equations of motions as follows:

$$Ma^n = P^n - F^n + H^n$$
$$a^n = M^{-1}(P^n - F^n + H^n)$$
$$v^{n+1/2} = v^{n-1/2} + a^n \Delta t^n$$
$$u^{n+1} = u^n + v^{n+1/2} \Delta t^{n+1/2}$$
$$\Delta t^{n+1/2} = \frac{(\Delta t^n + \Delta t^{n+1})}{2}$$
$$x^{n+1} = x^0 + u^{n+1}$$

Formula 10 - Central difference in LS-DYNA (35)

M denotes mass matrix, P forces on the system, F stress divergence vector and H resistance from hourglassing. The current time step is given by n, v is velocity, a is acceleration, u is displacement and x is the position of each node in the geometry.

Explicit time integration is not unconditionally stable, and the time step is bound for making sure it is stable for the solution. This is calculated by the following formula for shell elements, as can be seen in the LS-DYNA Theory manual (35):

$$\Delta t_e = \frac{L_s}{c}$$

Formula 11 - Time step size for shell elements (35)

where c is the sound speed given by:

$$c = \sqrt{\frac{E}{\rho(1-v^2)}}$$

Formula 12 - Speed of sound (35)

L_s denotes the characteristic length of the element, E, v and ρ are the common material properties. For shells L_s is given by formulae, which can be viewed in the LS-DYNA theory manual (35), for Hughes-Liu beam elements the formulae are similar, only that L_s is the element length.

For explanation of this bound Moan (32) state that *"When finding the maximum natural frequency of an element, one will see that the time step, Δt, must be short enough that information does not propagate across more than one element per time step."* It is taken on that the fastest information in these kinds of problems travel by the speed of sound.

Figure 13 is valuable for the understanding of the procedures in LS-DYNA. This shows how the solver works in each time step.
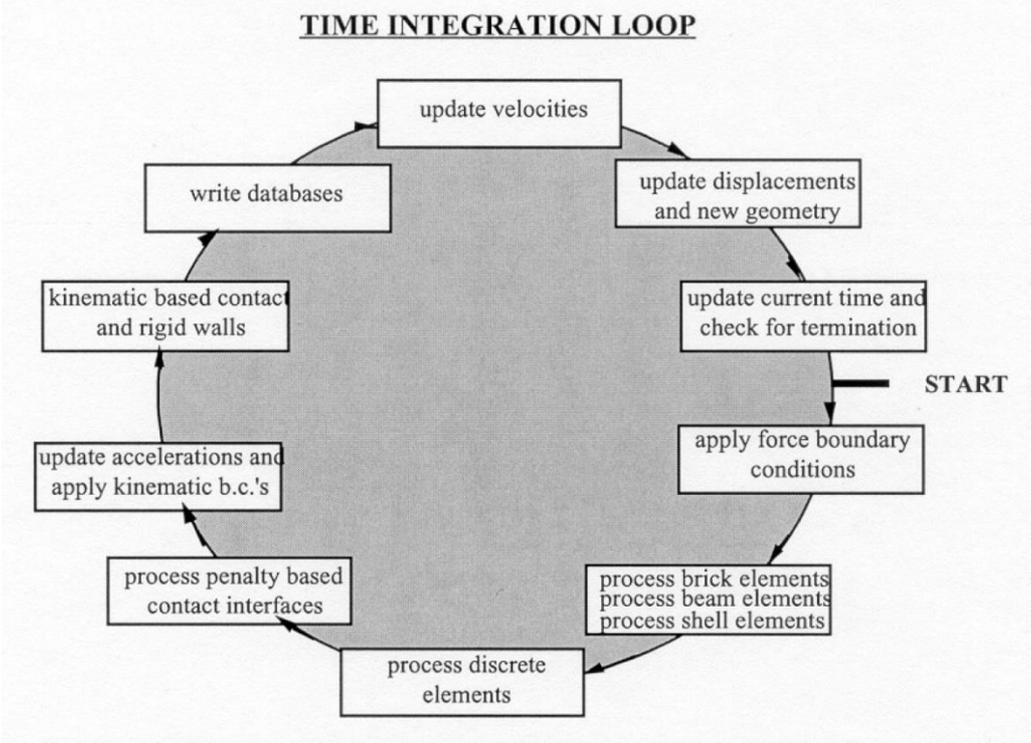
Figure 13 - LS-DYNA time integration (35)

4.2 ANALYTICAL METHODS

The simplified method is an analytical approach, with good applicability regarding ship collisions. This method is briefly discussed in section 5.1.1. A benefit of this method is the small need for computations to get an answer of relative accuracy.

For assessing a ship collision it is necessary to divide the side section into single structural parts, carry out the calculations for each and sum the forces. This is demonstrated by Haris and Amdahl in (47), and they show good agreements to numerical simulations. They conclude that for decisions in accidents and risk estimates their procedure might be the appropriate tool.

4.3 EMPIRICAL AND EXPERIMENTAL METHODS

By the use of collected data it is possible to derive formulae for ship collisions. As discussed in the introduction, Minorsky (12) uses empirical methods to estimate the

extent of deformations in ship collisions. This was done by systemizing data from well documented ship collisions.

The experimental investigations of full scale ship collisions are for obvious reasons materially very demanding, and therefore not commonly carried out. However there exist some studies of full scale experiments, model experiments as well as quasi static experiments on sections from the studied ship. These are convenient for benchmark testing where other methods can be approved by comparison. An example of full scale ship collision testing are carried out by TNO, which provide the basis for a study carried out by Konter et al. (48) for determining factors with importance in the nonlinear finite element models. Tabri et al. (19) carry out a series of experiments with model scale focusing on the dynamics of ship collisions. Quasi-static experiments on ship structure sections can be seen in a study carried out by Wang et al. (49). Here experiments are carried out on a structure similar to what can be found in a ship side or bottom. By the use of different indenters this series of test cover many bulb impacts and grounding cases. This study is further used for a numerical benchmark study carried out by Wu et al. (45).

Figure 14 - Quasi-static experimental test setup (49)

5 STRUCTURES

A ship can be seen as an advanced system of different structural members. On a basic level there are plates and beams. These make up panels, girders and intersections which in turn creates the structure as a whole. The aim of the first part in this section is to review studies of the structural members as well as to give an understanding of how the different members absorb energy. In the second part attention is given to some structural elements developed to increase the crashworthiness of a ship structure.

5.1 STRUCTURAL ELEMENTS IN A SHIP SECTION

Beams and plating are basic components in every structure. These can both withstand forces and moments, but their capability in doing so varies. A thin plate can by itself not take large bending moments; therefore it is often stiffened with stiffeners, i.e. beams. This combination is a stiffened plate, and in a ship it is used to transfer forces on the structure to the hull girder. The water pressure on the bottom of the hull is an example of such a force and the hull girder is the ship structure seen as beam. The stiffened plates transfer the forces to frames, made up by stiffened plate panels. In turn the frames carry the forces to longitudinal girders. As is easy to understand, when allowing for large deformations, as in the case of a collision, this complex system consists of very different elements with different capability to absorb energy. From this it can be deducted that the total energy absorbed in a ship's side during collision will depend on the place and angle of the introduction of the impact.

The purpose of this section is to obtain an understanding of the behaviour to be expected from each of these elements, thus being able to identify errors in the deformation pattern of the finite element modelling done in the analysis section of this thesis.

5.1.1 SIMPLIFIED METHODS

For a detailed background on simplified methods it is referred to the theory section of Hong and Amdahl (50), which is also conferred in the following description of simplified methods. Here only what is necessary for the basic understanding of the following formulae is reviewed.

Simplified methods commonly makes use of the following assumptions: a rigid-perfectly plastic material model, decoupled energy dissipation patterns, neglectable interaction

between structural elements and a simplification of bending deformation and displacement field. The kinematics are studied and both the mean and instantaneous crushing force can be developed. The study of the final deformation pattern is central in the development of such models.

One central concept important for the simplified methods is the plastic bending moment capacity for plane stress state, where t is the thickness of the component, and σ_0 is the constant flow stress:

$$M_0 = \frac{\sigma_0 t^2}{4}$$

<div align="center">Formula 13- Plastic bending moment (50)</div>

The effective crushing factor, denoted by λ, is introduced to take into account the fact that a theoretical structural fold cannot be completely compressed due to the material in the structure. One plastic fold is assumed to have a depth of 2H, and by deriving the energy (E_{total}) the following relation can be set up for the mean crushing force.

$$P_m = \frac{E_{total}}{2H \cdot \lambda}$$

<div align="center">Formula 14- Mean crushing force (50)</div>

5.1.2 ELEMENTS

Figure 15 illustrates plate field (green), web girder (red) and cruciform (blue) which are elements in a ship side section. They can all be described by the use of simplified methods and a brief description follow.
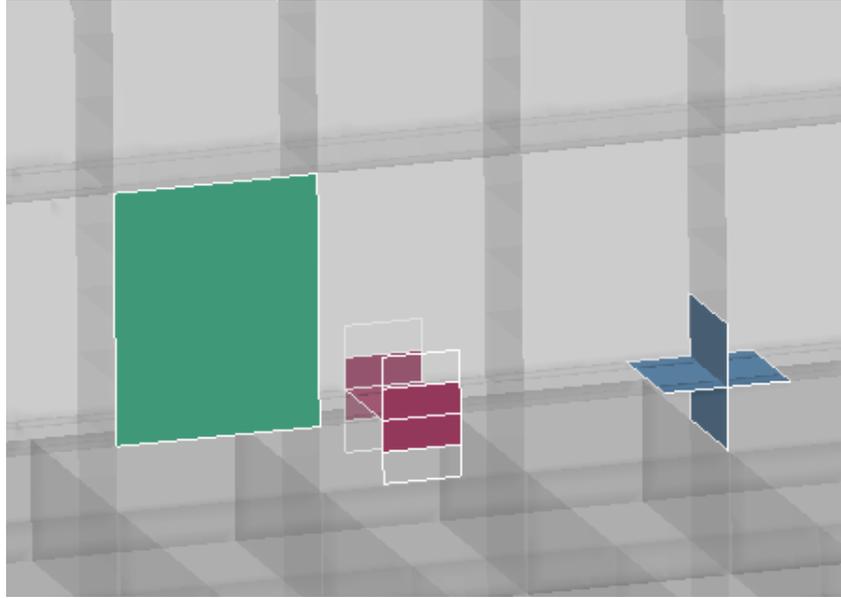
Figure 15- Structural elements in side section

Stiffened plate

For the resistance of a stiffened plate Haris and Amdahl (47) propose a formula with the geometry of the bow included. Their formula is cited in formula 15 and 16. Formula 15 is based on a formula proposed by Zhang (51).

$$P(\delta) = \frac{8}{3\sqrt{3}} \cdot \sigma_0 \left(t_{px} \cdot \frac{S_y}{S_x} \left(1 - \frac{\alpha\delta}{S_x} \right)^{-1/2} + t_{py} \cdot \frac{S_x}{S_y} \left(1 - \frac{\beta\delta}{S_y} \right)^{-1/2} \right) \delta$$

Formula 15- Formula for resistance of plate (47)

$$\delta_f = \left[1.316 \frac{S_x S_y}{\sqrt{S_x^2 + S_y^2}} \sqrt{\varepsilon_f} \right] \times \sqrt{\alpha}$$

Formula 16 - Failure displacement (47)

S_x and S_y denote lengths of the plate in x- and y-direction, respectively, t_{px} and t_{py} denote plate thicknesses with the stiffeners in x- and y-direction smeared according to the direction of the stiffener, σ_0 is the constant flow stress achieved by the average of the yield and ultimate engineering stress, δ is the central indentation of the plate, α and β denote the curvatures of the bow.

Web girder

Hong and Amdahl (50) study the different available formulae for crushing of web girders and propose their own formulae for this. For further details it is referred to their study. Here only the mean crushing force for a web girder with central load is cited as follows.

$$\frac{P_m}{M_0} = \frac{17.0}{\lambda}\left(\frac{b}{t}\right)^{1/3}$$

Formula 17 - Mean crushing force for web girder (50)

The total length of the girder is defined as 2b, meaning that b is the half-length, t is the thickness of the web girder and λ is the crush factor.

The number of folds in a girder is of the essence when making sure that a model is capable of describing the collapse. Formula 18 can be used to estimate the length of a structural fold, and figure 16 show the application of this.

$$H = 0.395b^{2/3}t^{1/3}$$

Formula 18 - Length of one structural fold (50)
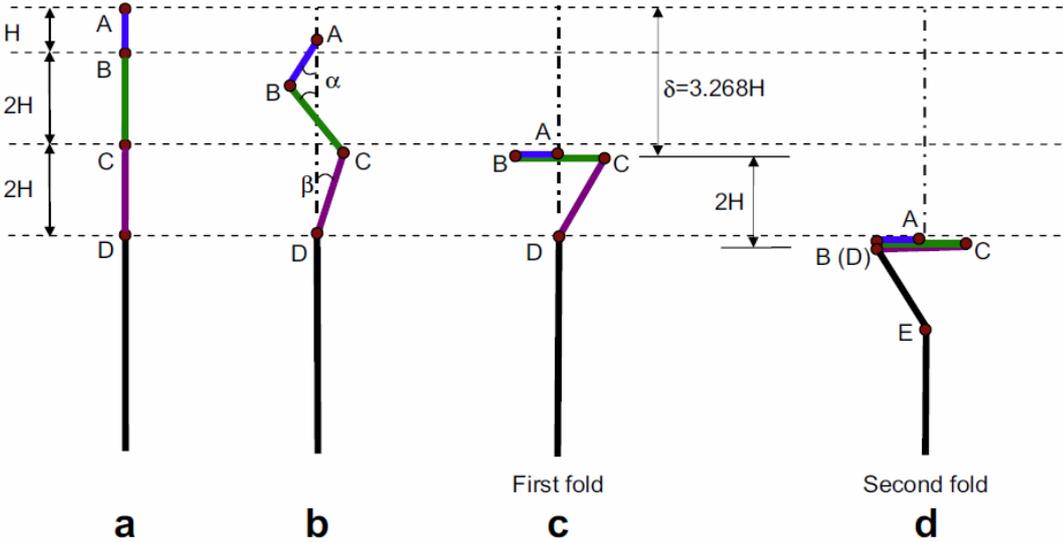
Figure 16 - Folding of web girder (central cross section) (50)

Cruciform

Haris and Amdahl (52) review analytical formulae from other authors, but do not present their own. Instead they study, among other, the behaviour of the cruciform by means of finite element method and propose an effective width of the cruciform. It is shown that the energy absorbed in an extent equal to half the height is dominant. The following formula is cited from (53) and provides analytical measures for the mean crushing force of a cruciform.

$$\frac{P_m}{M_0} = \frac{20.05}{\lambda}\left(\frac{C}{t}\right)^{1/2}$$

Formula 19 - Mean crushing force for cruciform (53)

C is the length of each of the four members of the cruciform, and t is the thickness. Figure 17 shows an example of the deformation pattern of a cruciform from the numerical studies of Haris and Amdahl (52).
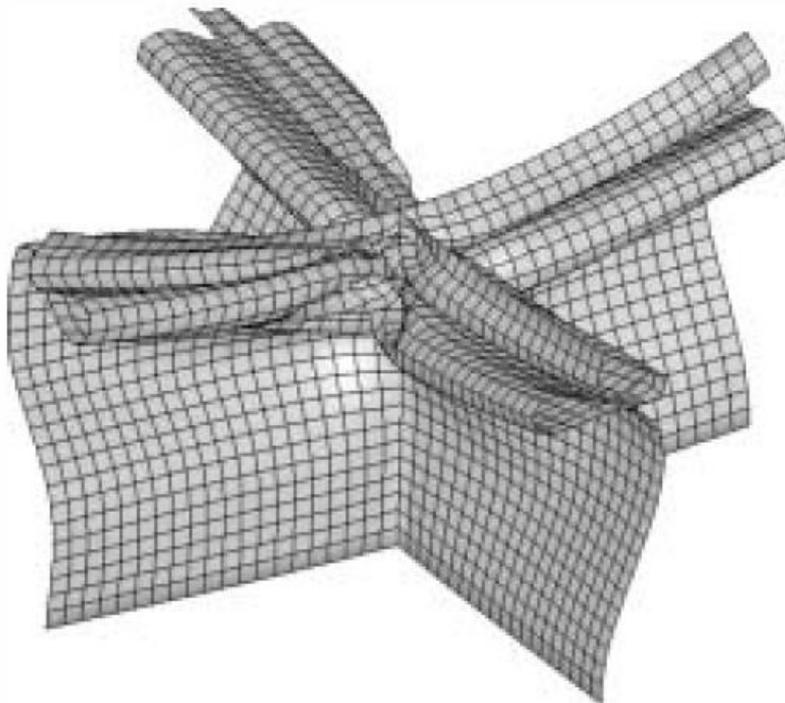
Figure 17 - Deformed cruciform (52)

5.1.3 SHIP SIDE

Here the procedure presented by Haris et al. (47) for analytical assessment between ships is followed to make a force indentation curve estimate for the ship section analysed in the chapter 6 of the current thesis.

Firstly the structure is divided in the elements as described. To illustrate this, figure 18 is cited from (47).
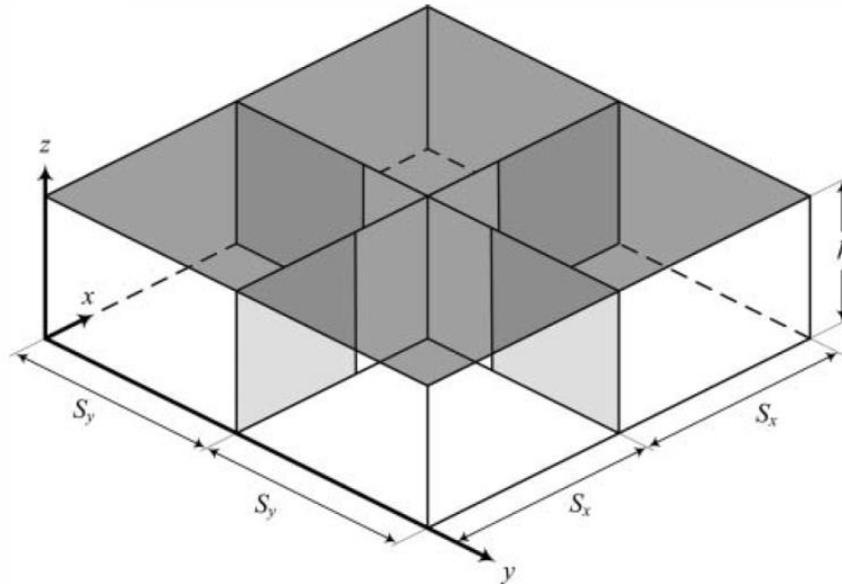
Figure 18 - Side structure (47)

The structure consists of four web girders, one cruciform and a plate, which is the outer skin. Due to the low contribution to the internal energy before the rupture of the outer skin the inner skin is neglected (47).

The formulae are the same as presented for each element. Table 2 shows the dimensions and the absorbed energy for these.

In the calculations only the behaviour of the section until rupture of the outer skin is considered. The bow is relatively small and does not hit any of the adjacent cruciform. Haris and Amdahl demonstrate the analysis of a collision where the bow hits the adjacent cruciform in (47).

Table 2 - Analytic estimate of absorbed energy in the different structural elements

Cruciform	C [mm]	t [mm]	Contribution	M0 [Nmm/mm]	Force [N]	Contributed Force [N]
	1200	10	0,25	11438	3441226	860307
	1200	10	0,5	11438	3441226	1720613
	1200	10	0,25	11438	3441226	860307
			Total contribution			3441226

Web girders	C [mm]	t [mm]	Contribution	M0 [Nmm/mm]	Force [N]	Contributed Force [N]
	1400	10	0,25	11438	916042	229011
	900	10	0,5	11438	790595	395298
	1400	10	0,25	11438	916042	229011
			Total contribution			853319

Shell plating		Sx [mm]	Sy [mm]	Tpx [mm]	Tpy [mm]
	1	2600	2100	11,8	10

As the forces in the shell plating depend on the indentation, these have not been tabulated in the same way as the others but are included in the force indentation curve as follows in figure 19.
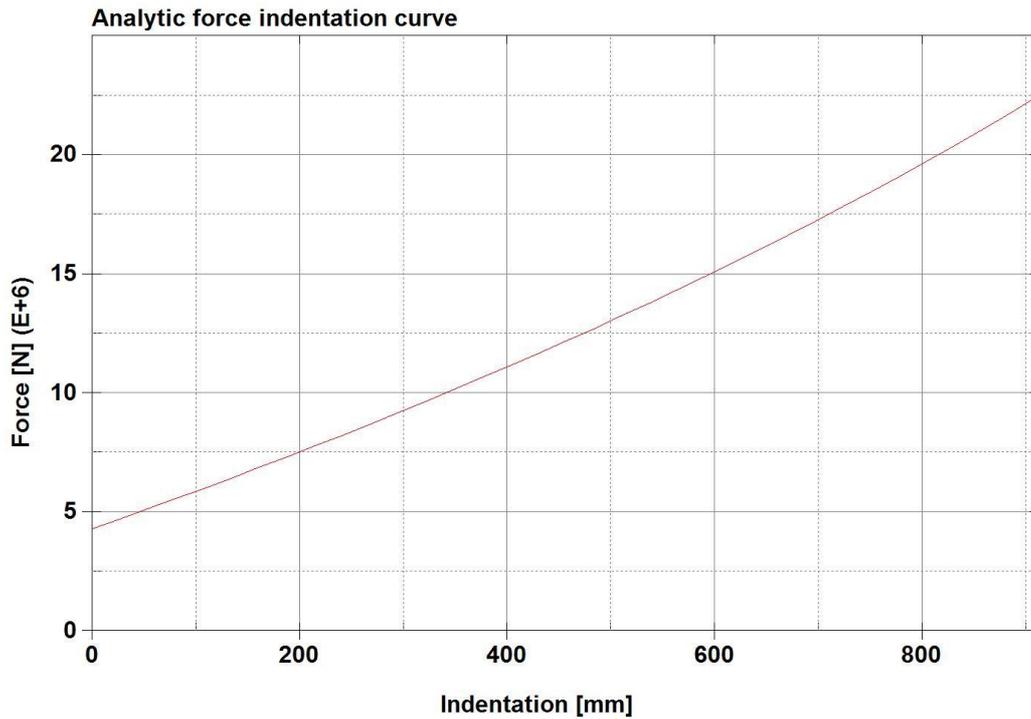
Figure 19 - Force indentation curve analytical

5.2 OPTIONS FOR IMPROVED ENERGY ABSORPTION

Focus on safety in all aspects of the maritime industry gives rise to research on solutions for increasing safety regarding ship collisions. One field of study is structures with high ability to absorb energy and there exist several proven concepts. The aim of this section is to briefly describe some of this research to give an indication to what can be expected for this type of structural element.

Hogström and Ringsberg (6) compares the structures in figure 20 namely; one standard ship side section (a), one section with a corrugated inner skin (b), one x-core (c) and one y-core (d).
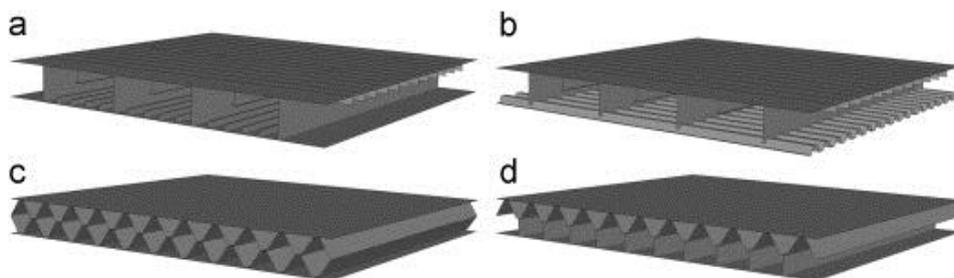
Figure 20 - Side sections (6)

For details it is referred to the paper as cited, as the scope here is to cite the outcome in terms of how capable these sections are of absorbing energy.

Behind the corrugated structure is the principle that increased indentation increases the absorbed energy, as energy is the force integrated over indentation. The core structures deploy a larger strength, i.e. a larger force, thus absorbing the energy over a decreased distance.

Figure 21 show an energy to indentation plot for the numerical results on experimental scale for each of these structures. These curves are the outcome of numerical studies in (6), where the setup is equal to what is found in the studies presented by Karlsson et al. (54), namely a 135mm rigid half sphere driven into the structures.
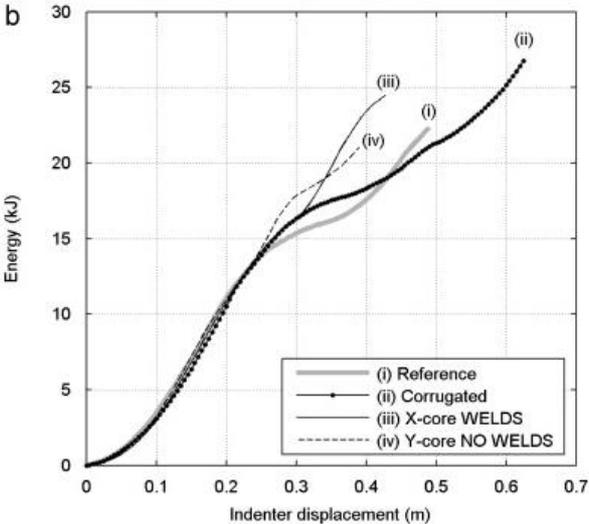
Figure 21 - Energy indentation curves for novel ship side structures (6)

An advantage of the x- and y-cores is the reduction of indentation to achieve a given amount of absorbed energy, while the corrugated solution makes use of increased indentation to absorb energy. It is observed that the corrugated concept absorbs the most energy, at an increased indentation, and the X-core absorbs more energy at a lower indentation.

Expense in the form of cost and weight is often of the essence for the evaluation of different solutions. This is assessed by Hogström et al. (6). They report the relative costs and weights for their experimental structures as follows:

	Relative Weight [%]	Relative cost [%]
Reference	0	0
Corrugated	-25	-23
X-core	+28	+19
Y-core	+1	-7

These might give an indication, but on full scale it is assumed that the reliability of these estimates depend on closer investigation, at least for the cost estimation. Costs are estimated for the two core structures by Klanac et al. (55) where the x-core structure is estimated to cost 1.8 times a reference structure, and 1.3 for the y-core structures. These numbers assume roughly the same weight for each of the sections and include folding, welding, handling and painting. In their study Klanac et al. (55) compare 10 different core structures, and conclude that the results in terms of energy absorption are promising. Ehlers et al. (7) present results from implementation of a core structure in a tanker and in a ROPAX vessel. They conclude that the final energy absorption is 30% better for the tanker and 50% better for the ROPAX vessel than for the reference structure.

6 MODEL

This chapter describes the model used in this thesis. It first presents the case of study, the simulation setup and model generation, then a verification study and lastly a description of the method used for the energy quantification.

During the modelling work several sources have been conferred, these include LS-DYNA Examples (56), LS-PrePost online documentation (4), LS-DYNA Keyword manual (46), the help function in MATLAB and the help function in PATRAN.

6.1 CASE DESCRIPTION

For the parametrical study the starting point is a standard cargo carrier. The ship section drawings can be seen in appendix A. Table 4 show the principle particulars and figure 22 shows an excerpt from the standard design, with the LNG tank included.

Table 4 - Principal particulars

Length over all	103.8m
Length between perpendiculars	101.6m
Breadth moulded	18.4m
Depth moulded main deck	9.05.
Draught, design	5.1m
Draught, scantling	5.3m
Maximum service speed	12.5 knots
Block coefficient	0.75
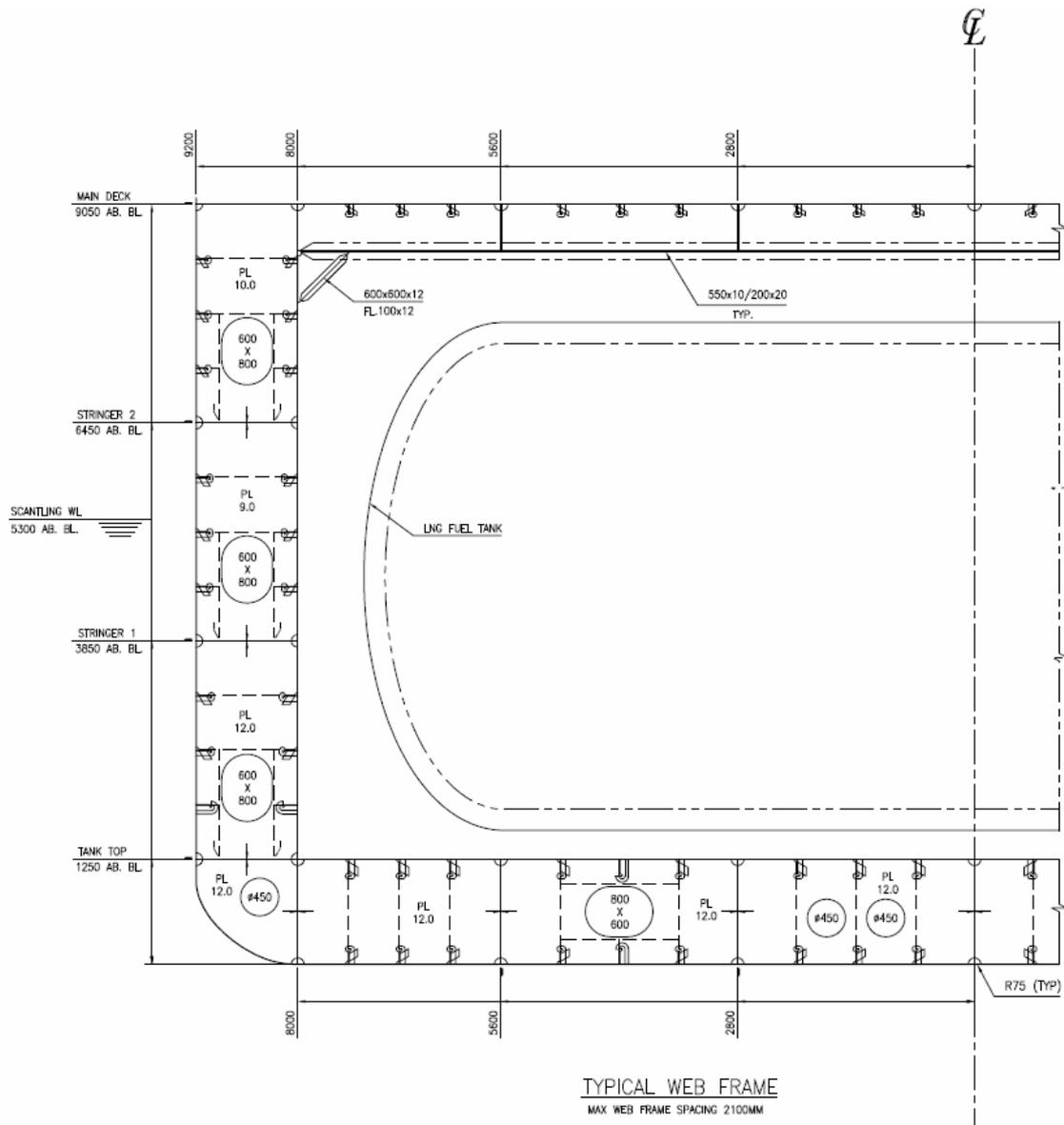
Figure 22- Cross section

Simplifications

The following simplifications are assumed, which do not comply with the drawings:

- All cut-outs other than the manholes are neglected.
- The bulb profiles are modelled as flatbars with the equivalent cross section surface area and height.
- All brackets in the normal frames are neglected.
- Stiffeners in proximity to the manholes are modelled across the manhole, such that all stiffeners on a plate field are directed in the same direction.
- The length of the hold is assumed to be 16800mm, corresponding to 8 webframe distances in the initial configuration.

- The gas tank is assumed to be enclosed in and isolated from the structural components in the ship side.

It is also assumed that, as only an increase of the structural parameters are considered, there are no other strength calculations according to rules that will be violated.

6.2 SETUP

Setting up a simulation such as this requires among other modelling of the geometry, meshing of this and definitions of the material model and boundary conditions. Parametrical studies require many nearly equal models. For this reason it is decided that a Matlab script is used for model generation. The simulation process outline is as follows:
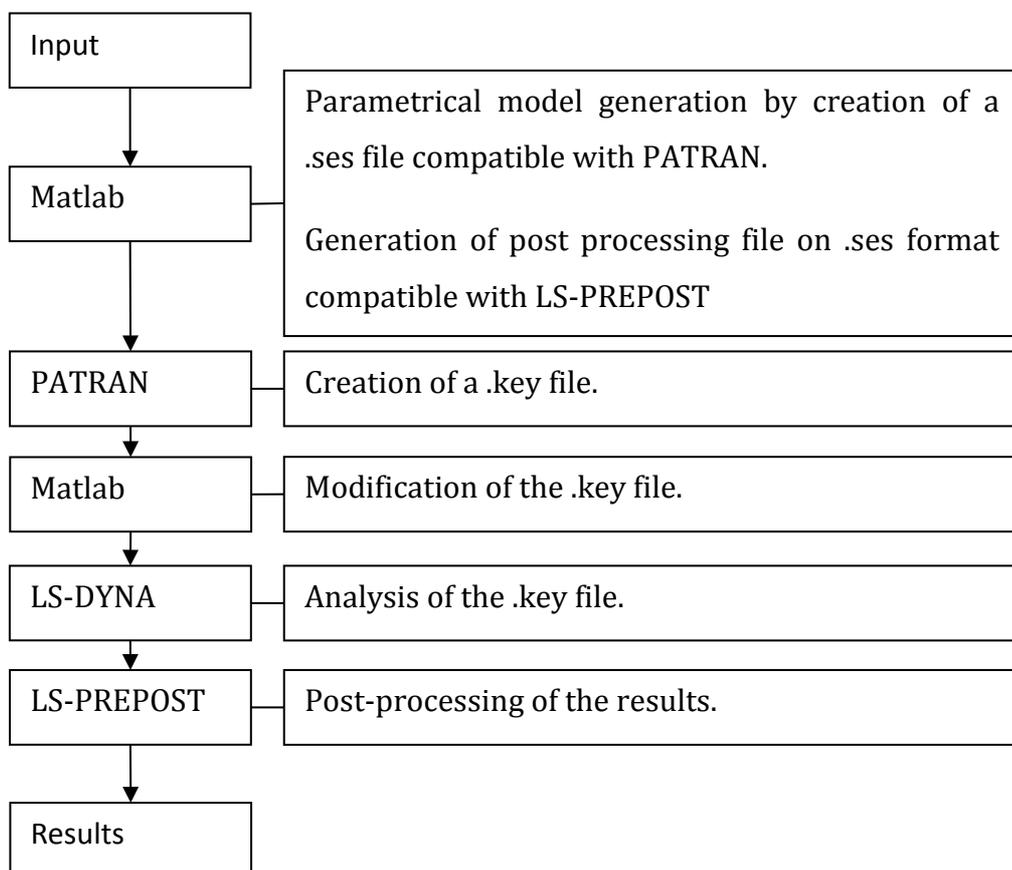
Figure 23 - Simulation outline

The simulation setup can be characterised as follows: the bulb, or indenter, is driven a given distance into the ship section at a constant speed and a right angle i.e. displacement control is used in the simulations. Force is obtained from the interaction between the bodies and it is then integrated over distance to give the energy. Following the advice given in Konter et

al. (48) the speed is set to 10m/s. Following the rules the distance of indentation is equal to the smallest allowable distance from the LNG fuel tank to the outer skin, i.e. B/5 (10) or 3.68 meters in this specific case. The boundary conditions for the bulb are that it is constrained against movement in all rotational degrees of freedom and against translation in the directions other than the striking direction. For the sidesection the boundary conditions are freely supported along all the edges at the location of the bulkheads, and free along the centreline of the ship section. Effects from the outer dynamics problem is disregarded, as it is only the struck ships ability to absorb energy that is under investigation, this means that uncoupling is used in the following.

Complete analyses input files are made by Matlab and PATRAN. Matlab is used to generate a session file for playback in PATRAN. This in turn generates a keyword file, which is used for the finished setup by means of Matlab. The resulting file is a LS-DYNA keyword file which includes all necessary commands.

6.2.1 BULB

Modelling the full bow structure is a time consuming task, and as this thesis is meant as a tendency study it was decided that the modelling effort should go into the ship side structure. Although it might alter the structures collapse pattern and ultimately the energy absorbed it was chosen to model the bow as a rigid cylinder with a half sphere at the end. This choice is based on the fact that it is very difficult to say anything exact about the bow structure of the striking ship. The top part of the bow could be modelled, and would probably account for some energy. As the stem can have a larger intrusion depth than the bulb before it would hit the LNG fuel tank it is left out of the current study. This is seen as a conservative assumption. From figure 6 it is seen that the addition of the upper bow structures might alter the collapse pattern in a favourable way, when accounting for the bow's deformations.

Ehlers et al. (42) use a rigid bow shape indenter. Zheng et al. (44) on the other hand makes use of a full bow structure which is treated as rigid. For cases where special collision scenarios are deemed very likely or on the post-accident study of a collision the striking structure might be closely considered as done by Hong et al. (16).

A Matlab function is made for the parametrical modelling of the bulb. Belytsckho-Lin-Tsay shell elements are used. Geometrical variables here are the radius in both ends of the cylinder and length of the bulb, as well as the offsets in x, y and z direction for easy positioning of the impact location. Figure 24 show the finished bow with the mesh used.
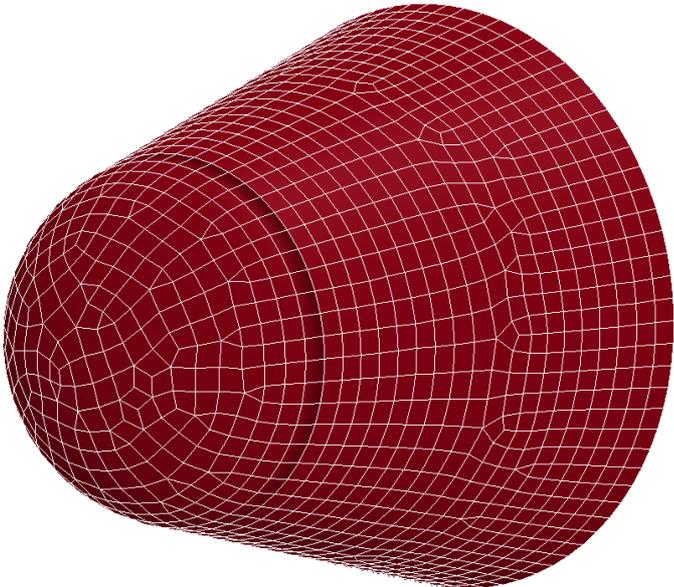
The size of the bulb needs to be in correlation to the collision scenario. This is to be a ship of approximately the same size hitting the ship at various impact points. The leading diameter is chosen to be 2.5 metres, the following is chosen to be 3.9 metres and the total length is chosen to be 4.0 metres. These measures are a downscaled version (the ratio between leading edge and taper is kept the same) of the bow used by Ehlers et al. (7). The measures are chosen to give a plausible bulb of a ship with comparable size to comply with the requirements as discussed in chapter 3 of the current thesis.

6.2.2 Ship section

The ship side is divided in three pieces, namely the double bottom, the deck and the sidesection. One Matlab function is made for each piece. All functions are parametric but the changes in the parameter study are made in the sidesection. Although Matlab is used for all calculations and the main model build up, the model language of PATRAN is used for the parametrical modelling and LS-DYNA is used for the analysis.

Table 5 show the parameters for the sidesection:

Table 5 – Variables in the sidesection

Table of variables in sidesection	Abbreviation
Length between webframes	lbwf
Height between stringers	hbs
Number of stiffeners in each stringer distance on outer skin	nssd_os
Number of stiffeners in each stringer distance on inner skin	nssd_is
Shell thickness of outer skin	stos
Shell thickness of inner skin	stis
Shell thickness of web frame	stwf
Shell thickness of stringers	stst
Stiffener thickness in outer skin	ssos
Stiffener thickness in web frame	sswf
Stiffener thickness in inner skin	ssis
Stiffener thickness in stringers	ssst

These variables are defined by means of an input file, and by running a Matlab script the model is built up to fit the input. The input file also includes the variables for the whole structure, but only the studied variables are listed in the table. An example of a finished side section is shown in figure 25.
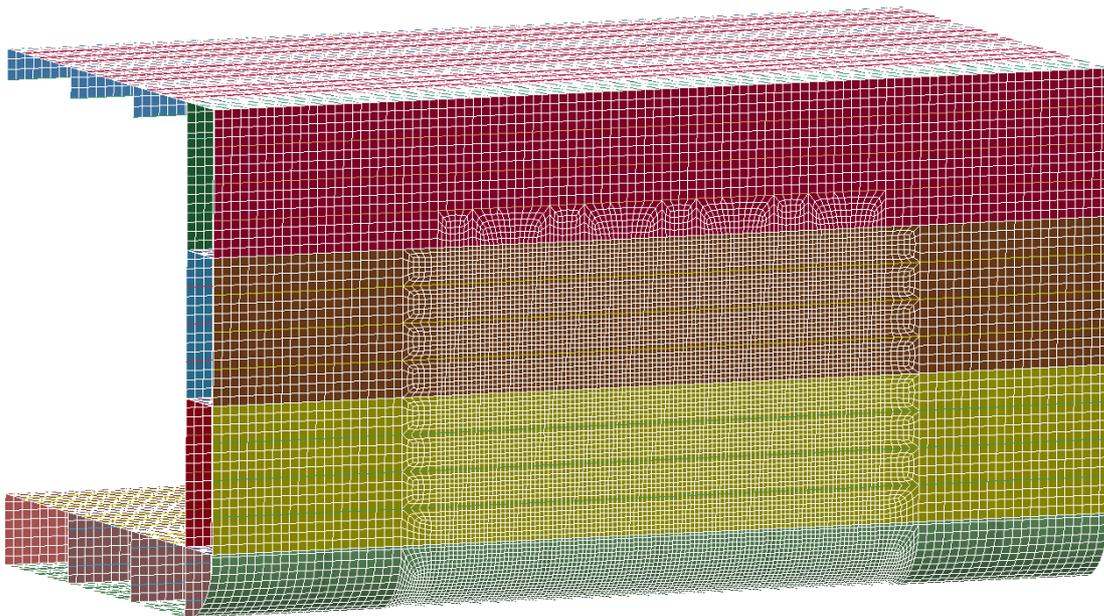
Figure 25 - Side section

6.2.3 PROGRAM DESCRIPTION

The main work in this thesis has been the programming done to create a parametrical model of the side section. This section aims at documenting the code in such a way that it is possible to understand it and to use it. For the detailed descriptions of variables and what each function does, it is referred to the appendix E, where the full code is given. Following is the program flow chart, starting at "runscript.m".
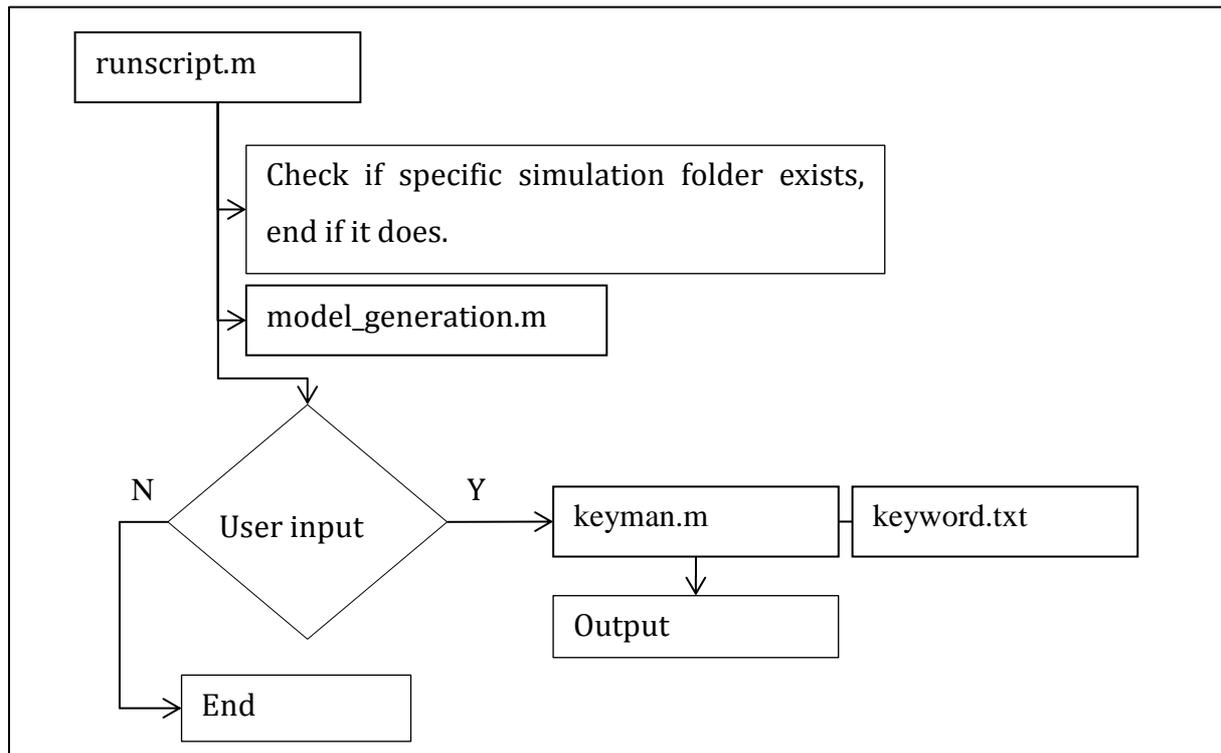
Figure 26 - Program flowchart overwiev

Input

There are three input files. The first named "keyword.txt" is dedicated to the keyword commands, which is the same for each simulation of the parameter study. It defines material data, indentation depth, simulation length etc. Secondly the file named "input.txt" is dedicated to the geometrical build-up of the model. Here dimensions and choice of sections are given. The sections are contained in the third input file, named "sectioninput.txt". This also defines the mesh sizes used. The start script is named "runscript.m" and other than being the governing script as shown in figure 26, it creates a map structure with folders for each bow position and the database. It also adds the input files to the database folder.

For the understanding of the program two other functions are deemed important to explain. Namely the functions already mentioned. Figure 27 and figure 28 show flowcharts for "model_generation.m" and "keyman.m" accordingly.
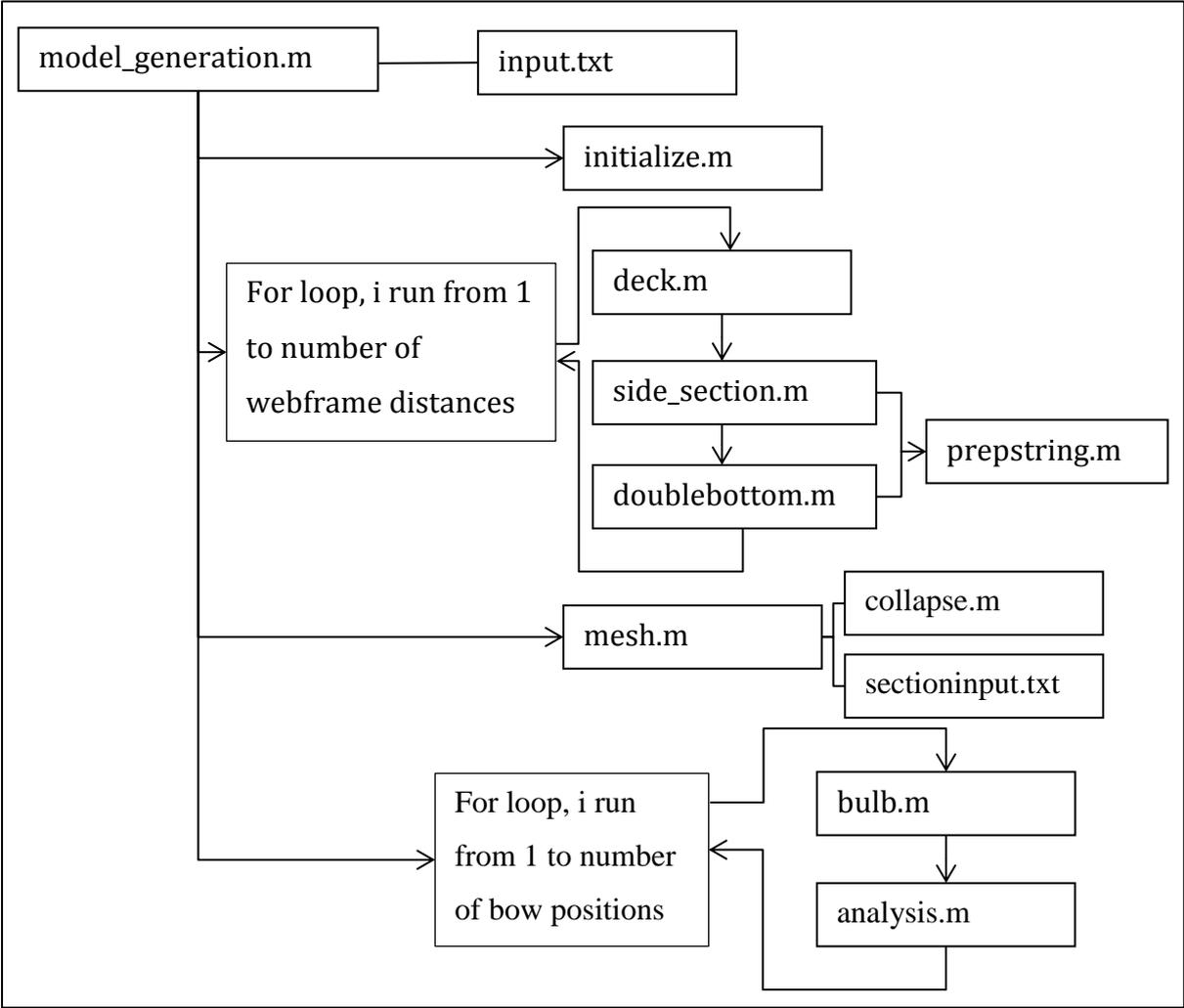
Figure 27 - Program flowchart, model generation
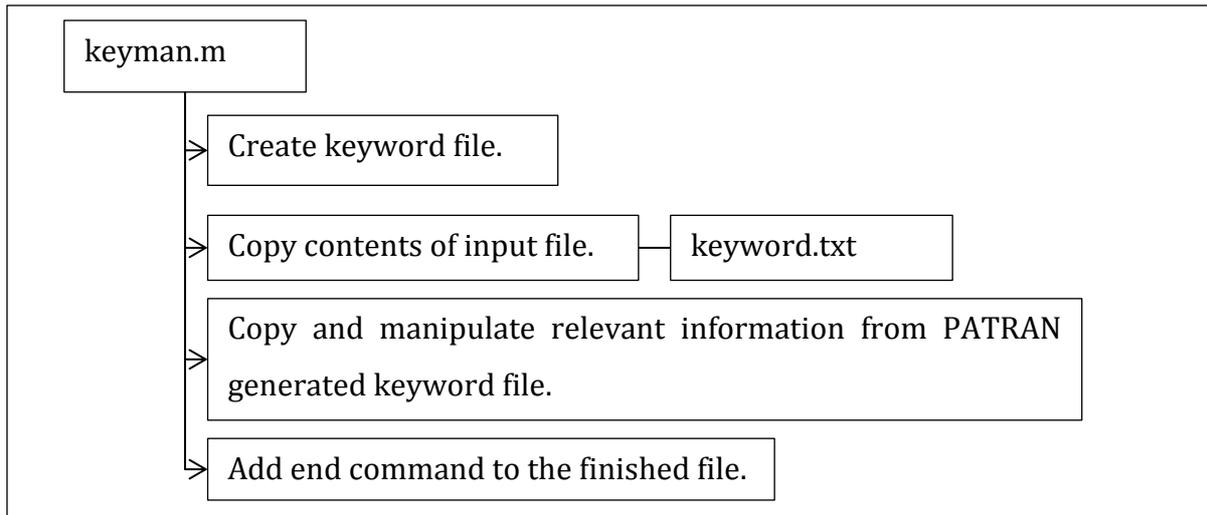
The files are, after successful execution of Matlab and PATRAN scripts, ready for analysis in LS-DYNA without any user-manipulation of the finished keyword files. An example of input files ready for execution in LS-DYNA is given in appendix B.

6.1.4 MATERIAL MODEL

Material data is taken as follows: Young's modulus is taken as 207 000Mpa, poisons ratio is taken as 0.3 and the specific weight is taken as 7850kg/m$^{3.}$ The bow is modelled as rigid by use of material 20 and the ship side is modelled by the use of material type 24. Information about the material types can be found in the LS-DYNA keyword manual (46). In material type 24, the plastic part of the stress strain relation is given as input, by the definition of a stress-plastic strain curve, as well as the standard material parameters. Numerical values are calculated by means of a Matlab script. The values and the curve as presented in chapter 4.1.3 of the current thesis are used.

The onset of fracture is determined by the curve in figure 12. For the element sizes used in the convergence study the following values have been read from this curve:

Table 6 - Failure strains used

Element length	Critical failure strain
100	0.27
80	0.29
50	0.33
30	0.38

Plate thicknesses are reduced according to the formulae in DNV class note 31-3 (27), as discussed in chapter 3 of the current thesis.

6.3 VERIFICATION

Simulations such as this are difficult to verify as the nonlinear behaviour makes analytical models challenging. Still the solution can be compared to analytical models, also convergence studies can be used as a verification that erroneous chosen factors in the model are of low influence on the results.

The force versus indentation and the energy versus indentation curves presented in the remainder of the current thesis have the force or energy on the ordinate and the indentation or the displacement of the bow into the side section on the abscissa. Also, zero indentation refers to the centreline of the outer skin, and the end of the curves refers to B/5, or 3680mm in this case.

6.3.1 CONVERGENCE STUDIES

Nonlinear finite element models require long computing time, making it is necessary to reduce the number of elements as low as possible without losing to much precision in the results. On some structural elements there exists recommendations on how big the elements should be or how many elements should be placed over a structural member. It is practical to use these as a starting point. Number of elements to be used for description of girder webs with linear response is recommended by DNV class notes (27) to be more than three. The number of elements needed to efficiently describe the failure of a half-length of a structural fold is recommended to be more than 8, according to Paik (57). In order to fulfil this requirement minimum 32 elements is needed in depth of the web frames and stringers in close proximity to the collision area, assuming that the minimum number of structural folds is two. This corresponds to an element size of maximum 37.5mm in this area for the initial setup. As the time step of a simulation is controlled by the size of the elements, not only is the size of the equation system controlled by the mesh size, also the number of solutions needed for a given problem depends on it. In turn this makes it essential to achieve the necessary precision with as large elements as possible.

For the convergence studies a collision position at the central web frame and at the lowest stringer is chosen. It is assumed that the results of this study are valid for other placements

of the bow, as long as the large deformations happen in the fine mesh region. It is also noted that the aim of the following study is the energy curves; these are obtained as the integrand of the force curves. Throughout the convergence studies the force curves are compared as the comparison of the energy curves would only make comparison more difficult. The shear factor in both beam elements as well as shell elements is set to 5/6, which is the value proposed in the LS-DYNA theory manual (35) and also used by Hughes et al. (36). In shell elements the number of through thickness integration points is 5, this is found to be the common practice. Hughes et al. (36) study the difference between 3 and 5 integration points, and for example Ehlers (8), Klanac et al. (58) and Hogström et al. (59) use 5 points. Figure 29 shows an example of the deformed side section.
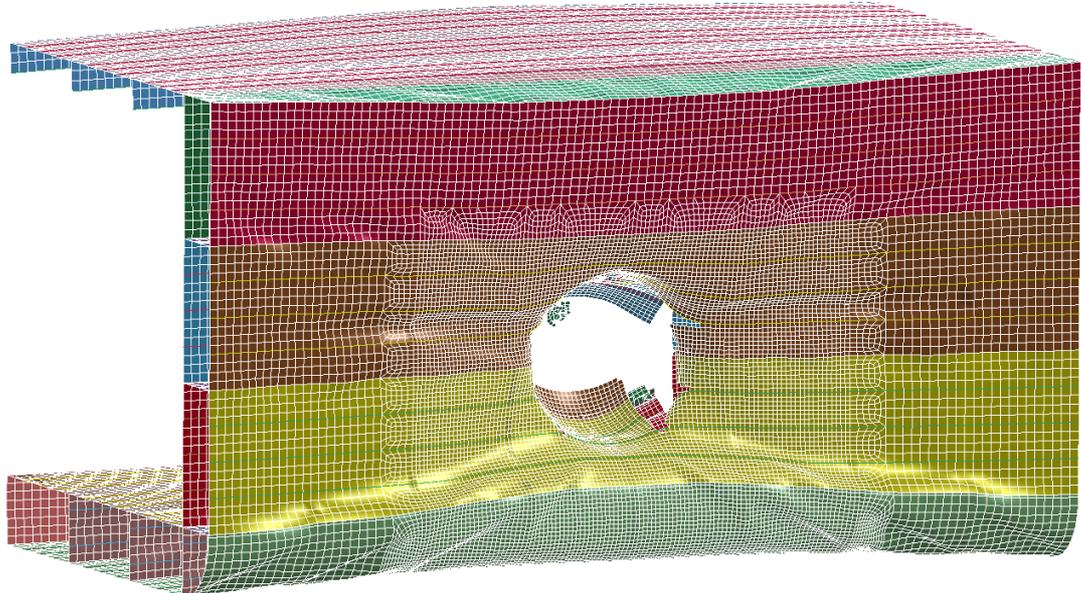
Figure 29 - Deformed side section

Fine mesh study

The scope of the first convergence test is to determine the impact on the force indentation curves from size of the elements in the collision area, hereafter referred to as fine area. Four different mesh sizes are tested, and these are 30mm, 50 mm, 80mm and 100mm. The mesh outside the studied region is set to 200mm. When changing the smallest mesh size, also the rupture strain must be changed and they are taken according to table 6. Figure 30 shows the comparison.
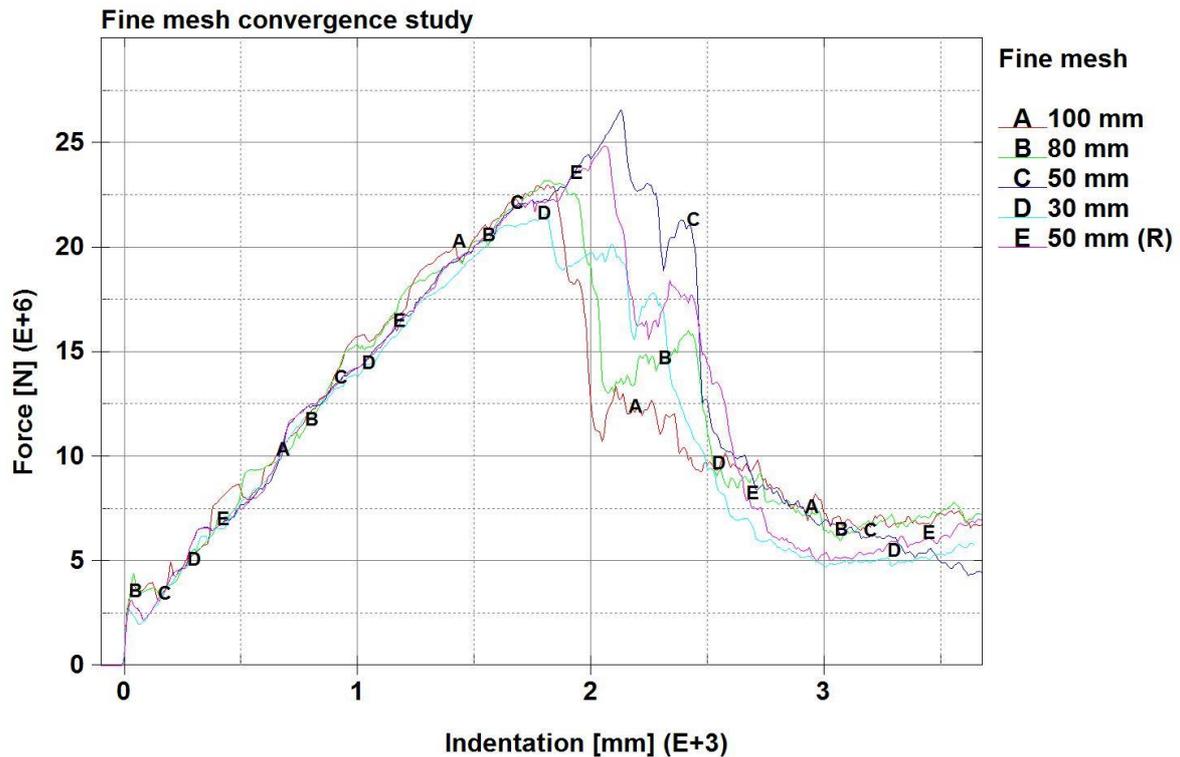
Figure 30 - Fine mesh convergence study

The trends of the curves correlate well, and an interesting feature is the significantly larger drops in force where rupture occurs in the cases where large elements are implemented. As the 50mm mesh lies above the others, another simulation is carried out. If this could be due to the failure strain, a significant decrease in the 50mm mesh curve should be observed when the failure strain is reduced from 0.33 to 0.325. This curve is denoted (R), reduced failure strain, in figure 30. It is decided that the reduction is observed and the deviation from the other curves is small enough to neglect for further consideration.

It is concluded that 80mm mesh size in the collision area is sufficient. This mesh gives a good description of the collapse at an affordable computational expense. It is noted that for studies where the aim is a precise estimation of collision energy of a specific case rather than a comparative study this should be reduced according to the number of elements per structural fold criteria.

Coarse mesh study

For the area not directly involved in the huge deformations arising from the collision, hereafter called the coarse area, the following mesh sizes are considered: 200mm, 160mm and 100mm. Figure 31 show the curves. Between the different meshes there exists good

correlation. It is decided that for the further studies 160mm mesh size in the coarse area is sufficient. This rest on the fact that it is observed an improvement of the smoothness of the mesh when using 160mm mesh instead of 200mm, i.e. the transition areas have a better mesh when 160mm is used.
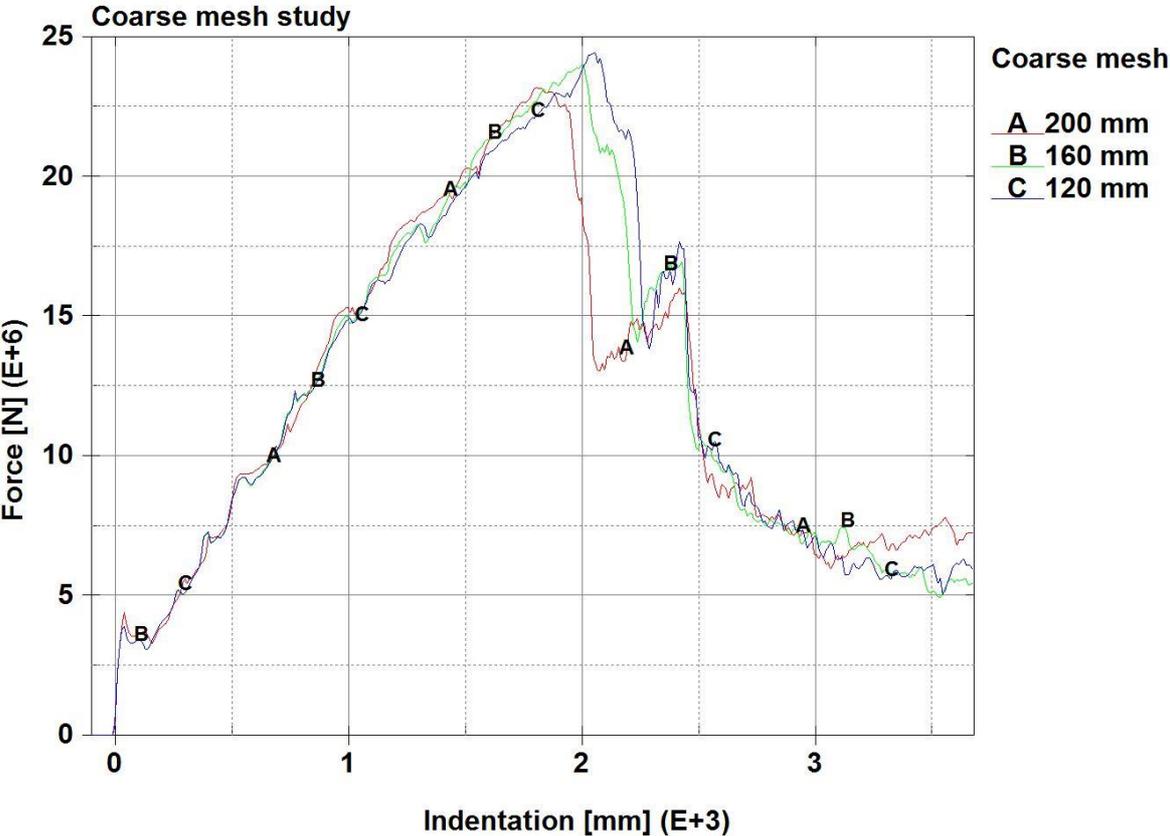
Figure 31 - Coarse mesh study

The final mesh sizes chosen are 160mm in the coarse area and 80mm in the fine area, and corresponding critical fracture strain value is chosen to be 0.29 according to Ehlers (8). When accounting for shell thicknesses between 10mm and 20mm the element length to thickness ratio is between 4 and 8 in the fine mesh region, which is found to be reasonable when conferring to what is used by Haris and Amdahl (52).

Velocity sensitivity test

The velocity of the indenter has been set to 10m/s according to the recommendations from Konter et.al (48). To investigate if there are significant errors in this assumption an analysis where the velocity is set to 5m/s is made. This analysis show only a small deviation, and as the scope of the following study is comparative, 10m/s is taken to be slow enough to ensure that quasi-static conditions is fulfilled. Figure 32 show the curves from the comparison.
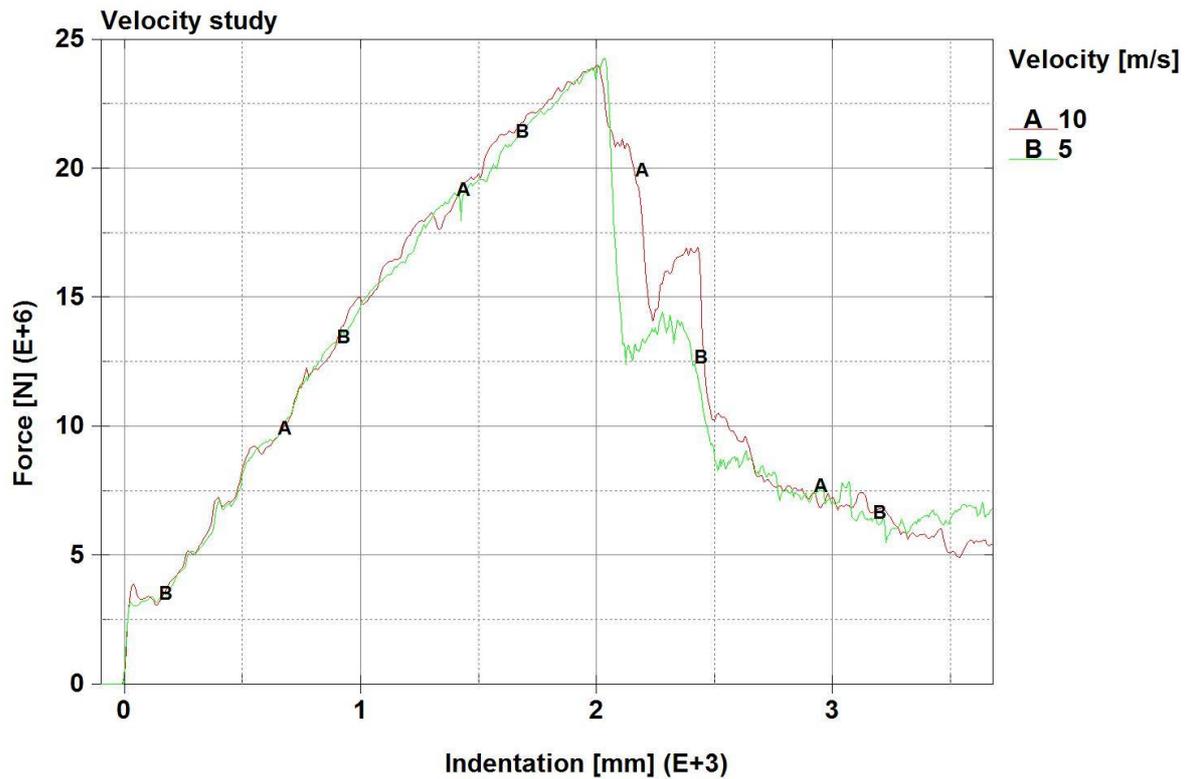
Figure 32 - Velocity study

Boundary conditions sensitivity check

Choice of boundary conditions is of the essence and different approaches are possible.

According to DNV class notes (27) and assuming that the section in consideration corresponds to tank hold type C, the following should be used. The full breadth of the ship should be modelled. In the length direction the model should consist of the hold of study as well as half of the adjacent holds. Different boundary conditions are to be applied at the bulkhead locations and at the middle cross section of the adjacent holds. This is, however, with regards to cargo tank analysis and linear analysis.

Other approaches include modelling the half section and constraining it only against translations at bulkheads as done by Ehlers in (8) and (20). Hong et.al (16) makes use of fully fastened boundary conditions at all boundaries, reasoning that the deformations will be local because of relative size and weight on the striking and struck vessel. In (16), the struck ship is a large FPSO and the striking is a supply vessel.

At this point three options are deemed possible. The first option is to keep all translational degrees of freedom fastened and the rotational free at the immediate bulkheads. The

second would be to include parts of the adjacent holds to account for the spring effect of this structure. The third option is to clamp all boundaries at the bulkheads. As an investigation of this three models are made. Figure 33 shows the outcome from the simulations.

Inspection of the strains at the constrained boundaries as well as the displacement of the free boundaries can give an indication of the precision of the boundary conditions. It is decided that only a half model is modelled in this study, as long as the values mentioned is within reasonable magnitudes.

The following boundary conditions have been studied:

BC type 1: Freely supported in the ends of the section of study, and fully clamped at the middle of the adjacent sections.

BC type 2: Freely supported in the ends of the section of study.

BC type 3: Fully clamped in in the ends of the section of study.
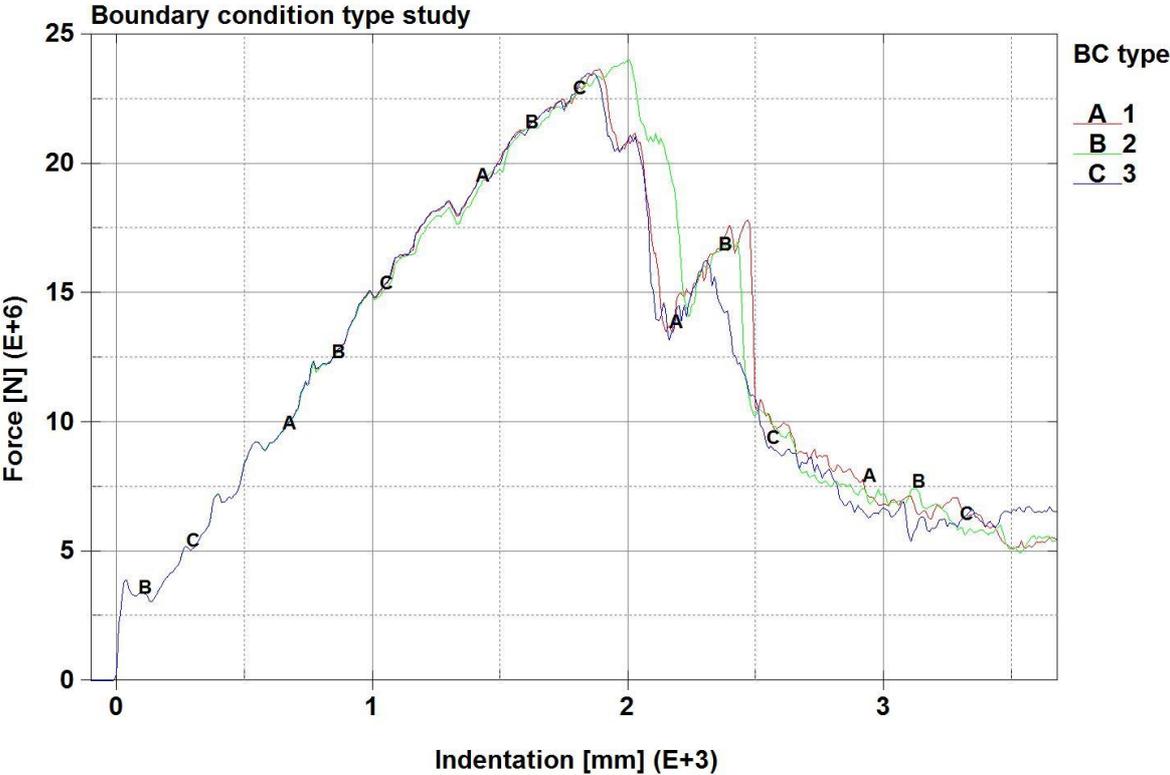
Figure 33 - Boundary conditions study

It is observed no excessive strains at the fixed boundaries; however, it is observed a significant elevation of the deck for the simulation where BC2 is implemented. This does not occur for BC1, which is assumed to be more accurate due to the modelling of parts of the adjacent holds. BC3 is observed to model this with certain accuracy, and is for this reason chosen for the boundary conditions. There are hardly any displacements in the depth direction of the section, this leads to the conclusion that a half section is sufficient for this study.

The boundary conditions were anticipated to have a large impact on the force indentation curves, this is however not the case in the current study. In the following parameter study a half section is modelled, and the only constraints are at the location of the bulkheads, where it is fully clamped.

Friction coefficient

There is no way of telling the exact condition of two ships colliding in the future, so the friction factors must be based on assumption. In the current study the values are assumed according to the ones used by Ehlers et al. in (42) namely a static friction factor of 0.3 and a dynamic friction factor of 0. Wu et al. (45) makes use of dynamic friction factor of 0.43 and 0.55 for static friction. This is, however correlated to the numerical simulations of a dry benchmark test with a polished indenter.

To investigate the impact on the solution from static and dynamic friction coefficient, simulations are made with different values implemented. The variation of the dynamic factor from 0 to 0.3 results in insignificant or no difference in the force. Figure 34 shows the results for the static friction variation study.
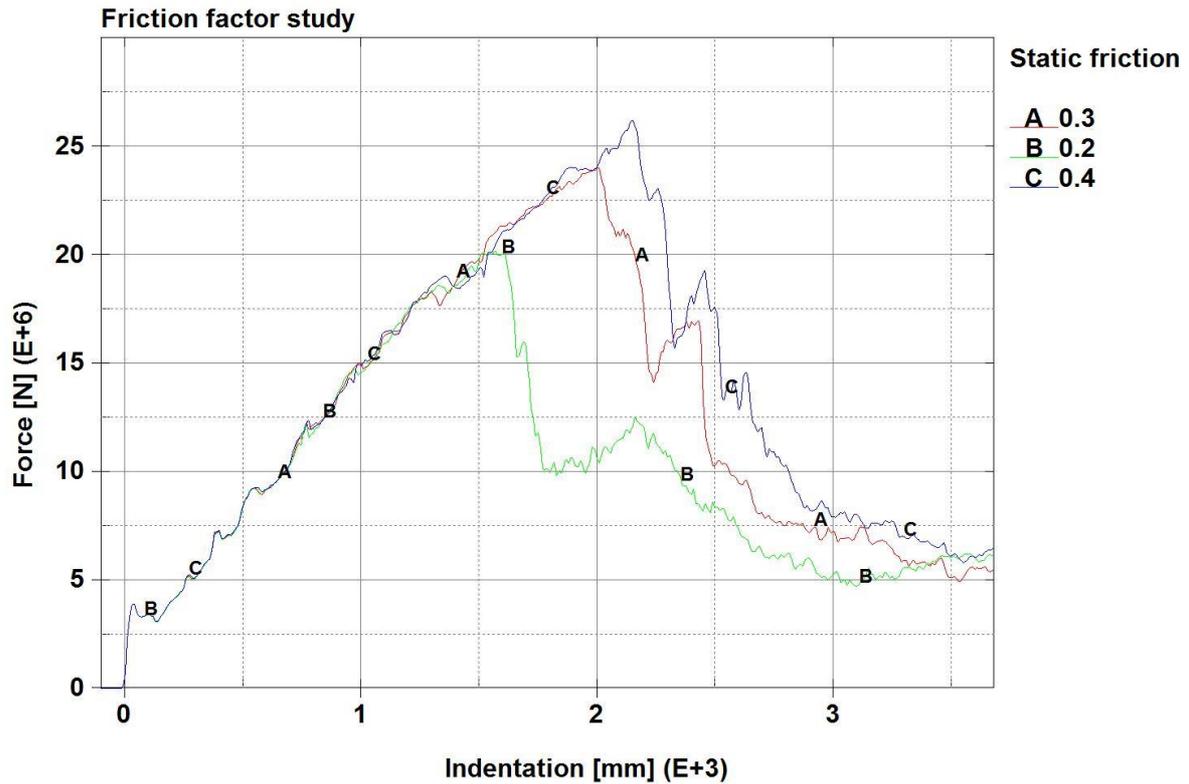
Figure 34 - Friction factor comparison

From the curves it is clear that the static friction factor is of great importance to the force, and this should be carefully chosen. In the rest of this thesis the values for friction is taken to be the same as used by Ehlers et al. (42) Static friction is set to 0.3 and the dynamic friction factor is set to 0.0.

Element type

As discussed in the chapter 4.1.1, the element type used, Belytscho-Lin-Tsay, sometimes encounter difficulties when it comes to warpage. To assess whether this is a problem in this given simulation, a model is made in which an improved element, Belytschko-Wong-Chong, is used. Figure 35 shows the result of this, and it is concluded Belytscho-Lin-Tsay elements can be used with confidence.
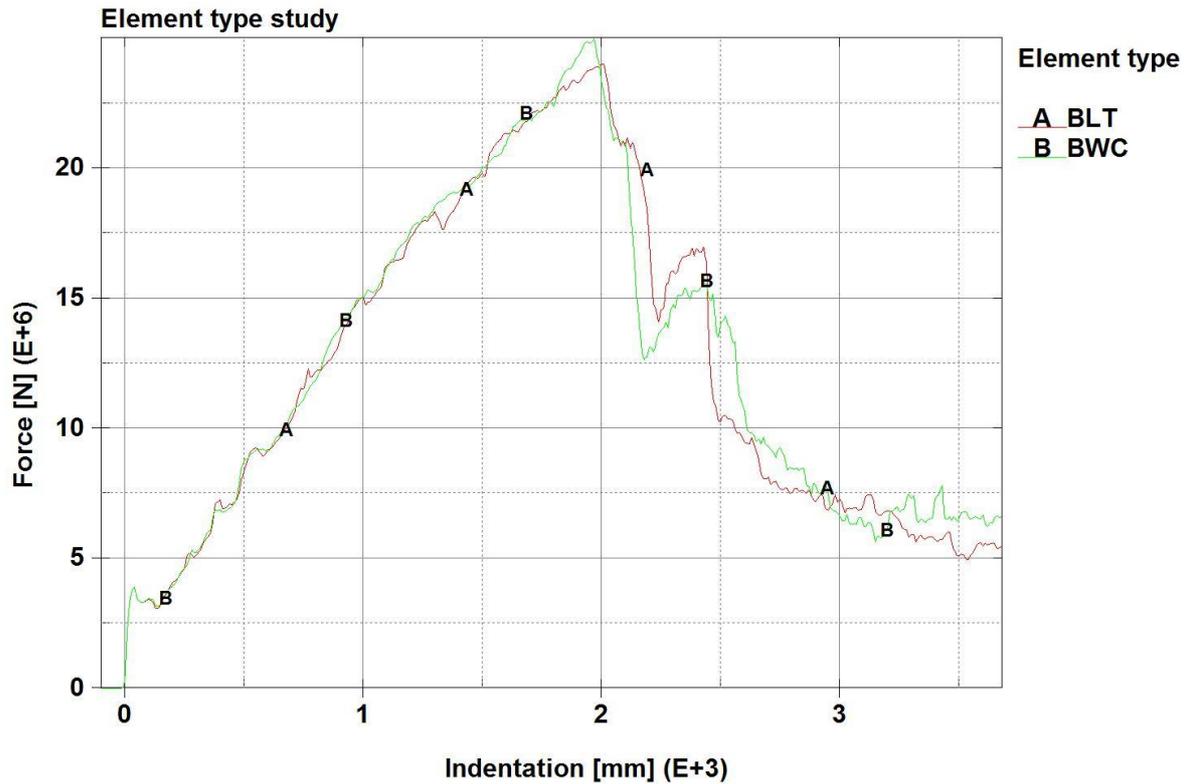
Element type study

Comparative study

The studies all show that different factors can make a huge impact on the measurements for collision energy. It is important to keep in mind that although these factors are of importance when it comes to the quantification of energy for a single case. They will to a certain degree be evened out when it comes to a comparative study. The goal of the following study is to assess the effect of changing structural arrangements, and other variables are the same for all simulations. Following the discussion above, Table 7 shows the values used for these variables.

Table 7 - Variables used in the finite element model

Coarse mesh size	160	mm
Fine mesh size	80	mm
Failure strain	0.29	-
Indenter velocity	10	m/s
Static friction coefficient	0.3	-
Dynamic friction coefficient	0	-

Other than the values in the table, the element type used is Belytschko-Lin-Tsay with 5 through thickness integration points. The shear factor is taken as 5/6. The boundary conditions are taken as clamped in the ends of the immediate hold, no constraints at the centreline. The modelling of half of the cross section of the ship is assumed to be sufficient for the following study.

It is observed that of the factors studied, the friction coefficient applied for static friction and the failure strain are of great importance on the results and should be chosen with care. For example the reduction of the velocity of the indenter and different boundary conditions are only of relative influence.

6.3.2 COMPARISON OF THE RESULTS TO ANALYTICAL METHODS

To control that the force measurements from the numerical model are plausible, the results are compared to values obtained by the use of analytical formulae following the procedure presented by Haris and Amdahl in (47). The method is discussed in chapter 5.1.3 of the current thesis and the analytical results are the same. For comparison one model was made with boundary conditions fully fastened around the structure of study as well as plate thicknesses and stiffeners to comply with the analytical model. The resulting comparison is presented in figure 36.
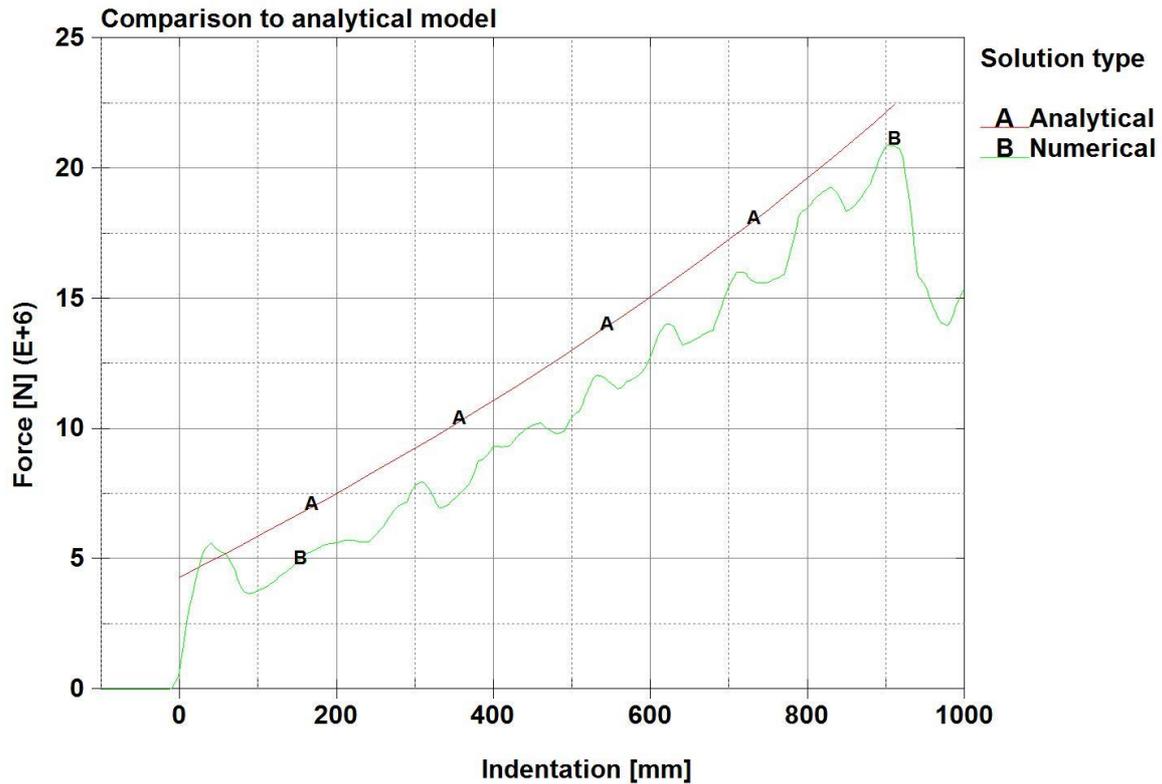
Figure 36 - Numerical to analytical comparison

It is seen that the analytical results generally lie over the numerical but compares to a certain degree. The point of fracture in the analytical model is found by setting it to the same as in the numerical model, so this does not give any indication if the comparison is good or not. From this comparison the force levels are seen as plausible.

6.4 SETUP FOR QUANTIFICATION OF COLLISION ENERGY

Precise information of a ship collision in the future is for obvious reasons normally not available. Therefore it is not practical to specialise the calculations for the quantification of collision energy. One variable anticipated having a huge impact on the damage pattern and ultimately the energy absorbed by the ship section is the striking position of the bow. Zhang et al. propose the use of a weighted function for a general quantification of the collision energy for comparison purposes (25). This is utilized by Ehlers in (8) for a particle swarm optimization of a ship side section with respect to crashworthiness. The setup defined by Ehlers is used in the preceding analyses, and the four collision cases are defined by the following:

1. The striking position is at the webframe in the middle between the tank top and the first stringer.

2. The striking position is at the intersection between the first stringer and the webframe.

3. The striking position is in the middle between web frames and at the height of the first stringer.

4. The striking position is in the middle between web frames and in the middle between the first and second stringer.

To use this procedure, four analyses are needed for each energy quantification simulation. The weighting factors used are generated according to Ehlers (8). During the analysis of the current study these need to be updated, as the number of stringers and webframes are intended variables. Following formulae apply:

$$W_{c1} = \frac{STR \cdot WF}{(2 \cdot STR + 1) \cdot (2 \cdot WF + 1)}$$

$$W_{c2} = \frac{(STR + 1) \cdot WF}{(2 \cdot STR + 1) \cdot (2 \cdot WF + 1)}$$

$$W_{c3} = \frac{(STR + 1) \cdot (WF + 1)}{(2 \cdot STR + 1) \cdot (2 \cdot WF + 1)}$$

$$W_{c4} = \frac{STR \cdot (WF + 1)}{(2 \cdot STR + 1) \cdot (2 \cdot WF + 1)}$$

Formula 20 - Weighting factors

Where WF is number of webframes between bulkheads and STR is number of stringers.

The weighting functions for the initial structural setup are: W_{c1} = 12/65, W_{c2} = 18/65, W_{c3} = 21/65 and W_{c4} = 14/65. And the following formula is used for the final energy measure, in accordance with Zhang et al. (25).

$$\overline{E} = W_{c1} \cdot E_1 + W_{c2} \cdot E_2 + W_{c3} \cdot E_3 + W_{c4} \cdot E_4$$

Formula 21 - Final energy

E is the collision energy measured for each collision case.

In the following study this weighting is carried out for each simulation, so that the presented curves present the final energy measure for all indentations up to max indentation.

6.4.1 FINISHED MODEL FOR ENERGY QUANTIFICATION

With the results from the sections describing convergence etc. it is now possible to set up the initial energy quantification model. As earlier mentioned it consists of four simulations. At this stage the reductions of the plate thicknesses for manhole cut-outs are implemented. Figure 37 show the final energy indentation curve for the initial setting of the parameter study, the individual force indentation curves for each bow position is found in Appendix C.
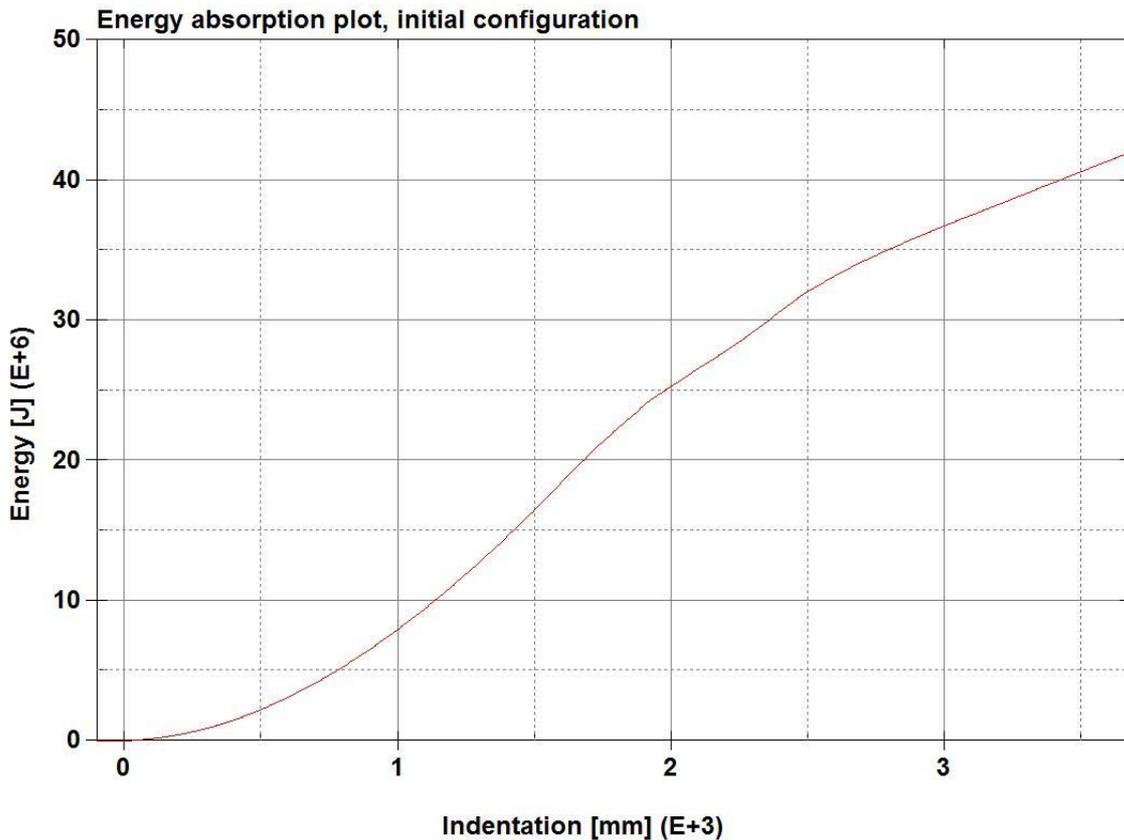
Figure 37 - Initial energy displacement curve

6.4.2 CRITERIA FOR EVALUATION

There exist different criteria when assessing structural arrangements for crashworthiness. Hogström et al. (6) evaluates four different innovative side sections on the following criteria: the intrusion depth when the colliding ship is at rest, the amount of dissipated energy at the fracture of the inner skin, the amount of dissipated energy when the colliding ship is at rest, size of the damage on inner skin, weight of the sections as well as the cost.

Ehlers (8) makes use of an energy to mass ratio, defined as the energy dissipated at the fracture of inner skin divided by the mass.

In their concept procedure regarding alternative arrangements for maintaining the safety with respect to damage stability Zhang et al. (25) use the rupture of inner skin as criteria.

In a submission to IMO by GL, Janse (26) propose a procedure for assessing the safety in the collision case. As earlier discussed, the *"equivalent crashworthiness approach"* is one of two possibilities for allowing closer positioning of gas fuel tanks. This document gives a procedure, and examples of the use of this, which follows the same principles as found in the alternative arrangement concept by Zhang et al. (25). On the other hand, it does use another failure criterion; namely the impact of the fuel tank. And the acceptance criterion is the absorption of the same amount of energy at failure for the modified and initial design.

In the following study the failure criteria follows the one used by Janse (26). The comparison criteria used in the following analyses are; the reduction of the indentation at which the equivalent energy is reached divided by the amount of steel added. Equivalent energy is the energy dissipated by reference, or initial, structure at the indentation where the bow would hit the fuel tank. By this it is assumed that the bow is the first member to strike the tank, meaning that it breaches through the inner skin and all structural members are folded away.

The steps followed to obtain the evaluation comparison ratio R during the analysis work are:

1. Create energy indentation curve for a reference design.
2. Read the value E_{eqv}, the energy absorbed at the indentation where the tank is impacted, D_{tl}.
3. Create energy indentation curve for the new design.
4. Read the indentation at which the E_{eqv} is reached, D_{eqv}. D_{eqv} is called safe distance in the following.
5. Find the added weight, M, from the models.
6. Use the following formula for obtaining the comparison ratio:

$$R = \frac{D_{eqv} - D_{tl}}{M}$$

Formula 22 - Calculation of comparison factor

Figure 38 show this principle, here the strengthened design is the initial energy absorption curve scaled by a factor of 1.5 for the demonstration.
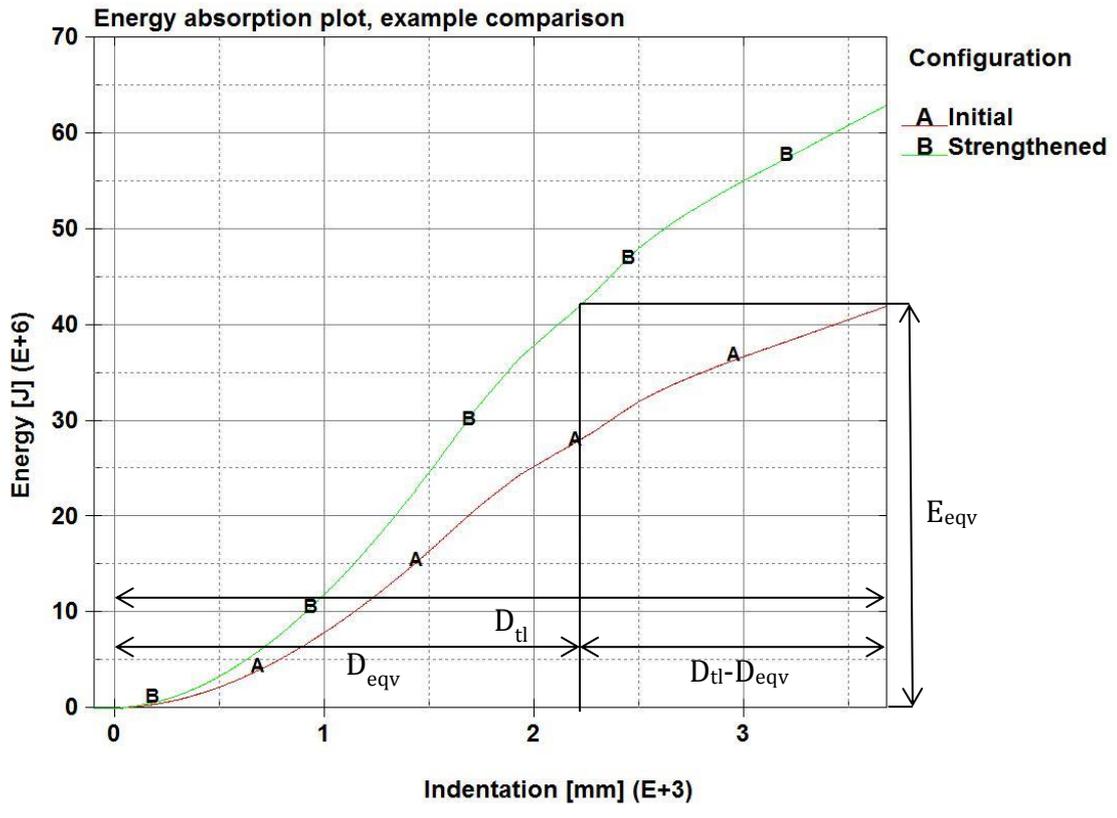
Figure 38 - Notation for comparison ratio

7 PARAMETER STUDY

This section presents the results of the simulations made in this thesis. First an examination of collision absorbance capabilities for designs with ice strengthening implemented is presented, secondly each of the structural parameters are varied, and lastly some of the structural parameters are investigated with parallel variation.

Note: In the following "safe distance" is often used and in this thesis this is defined as the distance by which the energy absorption is the equivalent of the absorbed energy at full indentation of the initial condition. This is shown by D_{eqv} in figure 38.

7.1 EXAMINATION OF THE INCREASE IN ENERGY ABSORPTION BY ICE-CLASS

Classifications for operation in ice are today commonly carried out. Strengthening for ice conditions often uses the yield criterion, i.e. negligible deformations from impact. Collision is considered to be an accidental event and large deformations occur. Commonly other failure criteria are used, but in the current study it is the impact of the fuel tank. The reinforcement required for ice navigation also gives stiffening in the collision case. To investigate the effect of the ice stiffening in the collision case two models are made according to drawings, found in appendix A. These designs are the same as the section previously studied; the only change is the implementation of ice strengthening according to DNV ICE-1A (60) in two different manners. Both models utilize an increase in plate thickness in the region where ice is a problem. The first makes use of increased numbers and sizes of the longitudinal stiffeners, whereas the second implements transverse ice frames, where the stiffeners are vertically directed. The second implement slightly smaller plate thickness in the ice belt. It is also noted that the material in the ice belt is of DNV grade B steel, which is a better quality than grade A, which is the steel quality used elsewhere. In this study all material is taken as described in the previous material section, this is taken as a conservative assumption as grade A is the steel of lower quality. Following is the resulting graph showing the different energy absorption curves.
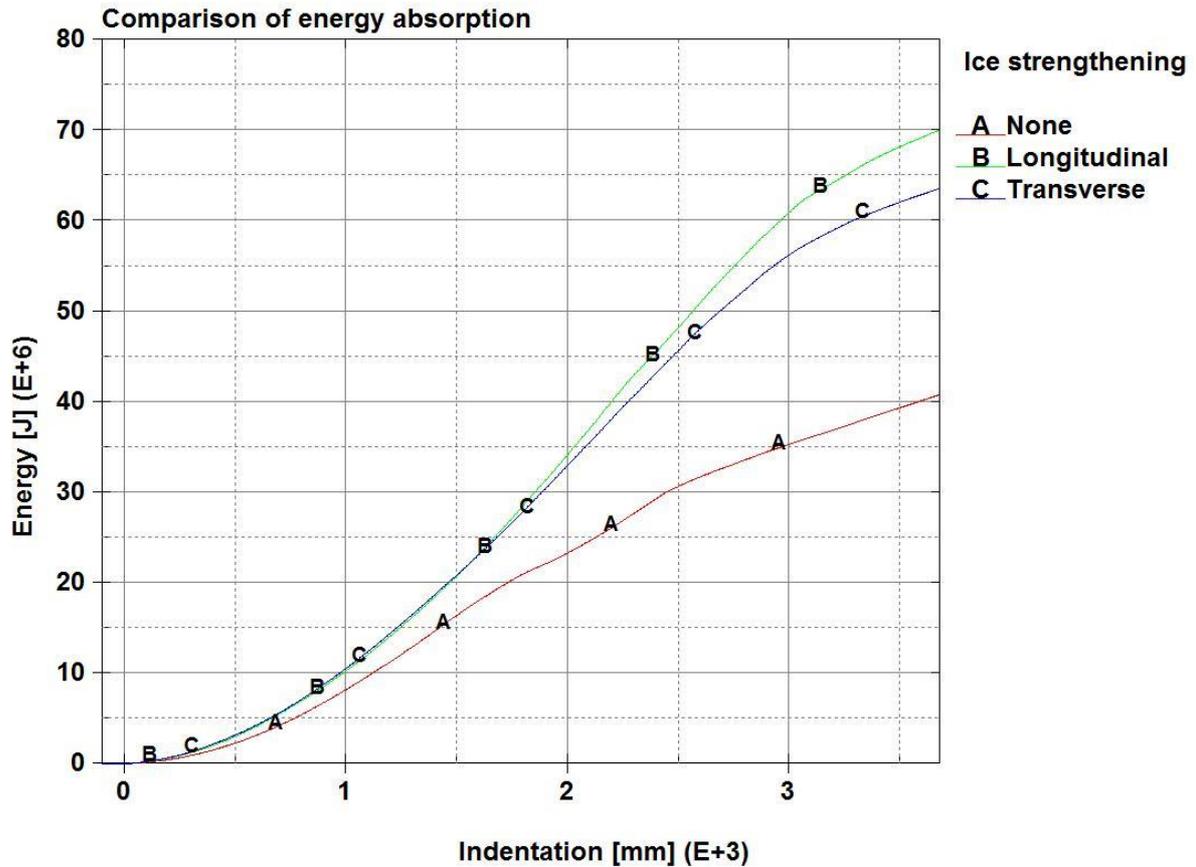
Figure 39 - Effect of ice stiffeners

It is seen from the curves that the longitudinal stiffening proves to be the most energy absorbing solution, and it absorbs the equivalent amount of energy from the initial setting at full indentation (3.68m) at 2.23m whereas the transversal stiffened design absorbs it at 2.31m. Following the equivalent or better safety principle, it is safe to reduce the minimum distance from the outer shell to the fuel tank to the given measures. Table 8 sums the results and provides a measure for comparison, namely the reduction in safe distance per ton steel added for stiffening [m/ton].

Table 8 - Comparison of ice stiffened designs

Stiffening	Reduction in minimum distance [m]	Mass of half section [ton]	Increase in steelweight [ton]	Ratio [m/ton]
None	0	96.01	0	-
Longitudinal	1.45	101.6	5.01	0.2895
Transverse	1.37	101.7	5.07	0.2704

7.2 INITIAL STUDY

Each structural parameter is varied from the initial value and with 50% and 100% increase of the initial quantity of steel; this does however not apply for the webframe study. In the following study some of the variables from the drawings are replaced with the variables listed in table 9. Thereafter only the parameter of study is varied.

Variables, maximum and minimum of these are listed, the full parameter matrix can be seen in appendix D.

Table 9 - Variables in the parameterstudy

Variable	Abbr.	Initial	Min	Max	
Length between webframes	lbwf	2100	1400	2100	mm
Height between stringers	hbs	2600	1560	2600	mm
Number of stiffeners in each stringer distance on outer skin	nssd_os	3	3	7	-
Number of stiffeners in each stringer distance on inner skin	nssd_is	3	3	7	-
Shell thickness of outer skin	stos	10	10	20	mm
Shell thickness of inner skin	stis	8	8	16	mm
Shell thickness of web frame	stwf	10	10	20	mm
Shell thickness of stringers	stst	10	10	20	mm
Stiffener thickness in outer skin	ssos	10	10	20	mm
Stiffener thickness in inner skin	ssis	10	10	20	mm
Stiffener thickness in web frame	sswf	10	10	20	mm
Stiffener thickness in stringers	ssst	10	10	20	mm

7.3 PRESENTATION OF RESULTS

Energy absorbance comparisons for each variable follow, the increase in weight and the resulting decrease in safe distances is summed in table 10, following the graphs. Chapter 8 compares and makes use of and places the results obtained in a bigger picture.

Remark: (R) denotes that the failure strain have been adjusted to account for deviations in the element size. This is done by linear approximation, as the smallest element sizes deviate only slightly.

Maxweight configuration

To set an upper boundary for the solution space, one configuration where all input is given to add as much steel as possible is made. Figure 40 shows the initial condition compared to the maxweight condition. This describes the lower and upper boundaries of the solution space in which all solutions in this identification study should lie within.
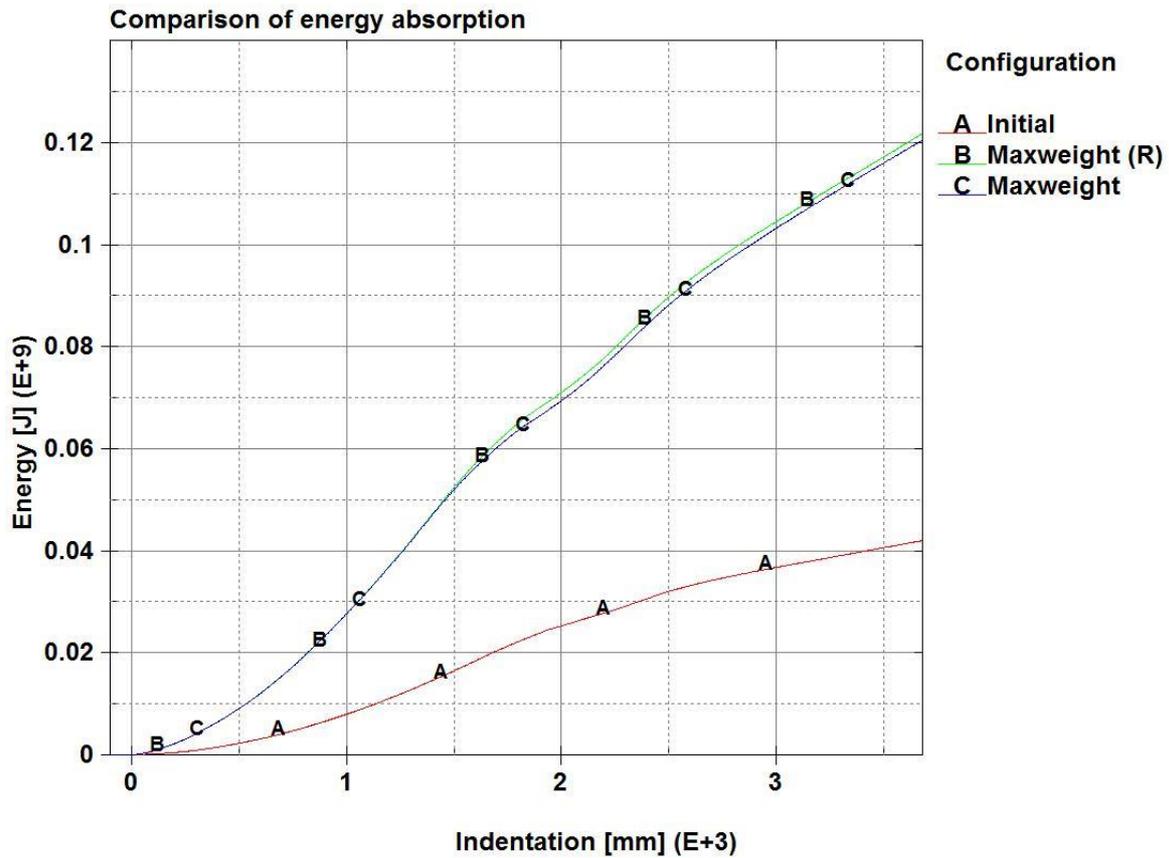
Figure 40 - Maxweight configuration study

It is observed that by increasing all variables, naturally there is a large reduction of the safe distance. For comparison to Minorsky's method (12) the weight of the impacted part of the maxweight section is in the affected region (only the double side structure) 98.36 tonne and the initial section weighs 32.22 tonne giving a weight ratio of 3.05. The ratio for max energy absorption is 2.87, this is in reasonable correlation to Minorsky's formula where the amount of deformed steel is proportional to the energy absorption. Here the increase in mass is almost proportional to the increase in energy absorption.

7.3.1 WEBFRAME PARAMETERS
Number in each tank hold
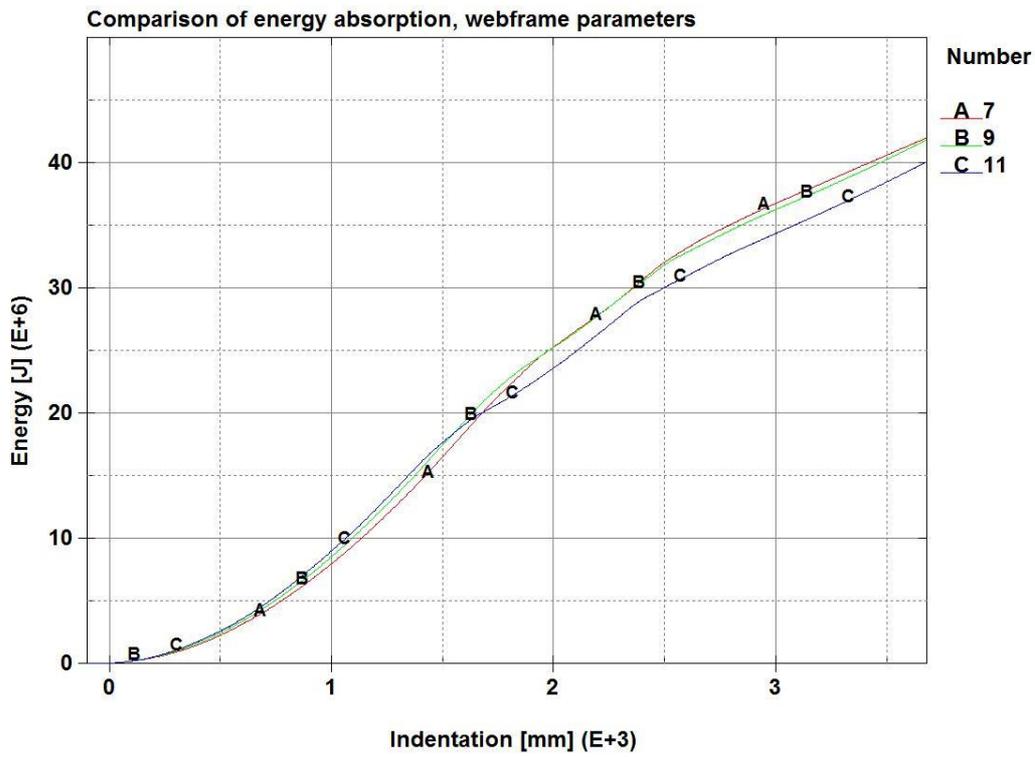
Figure 41 - Number of webframes study
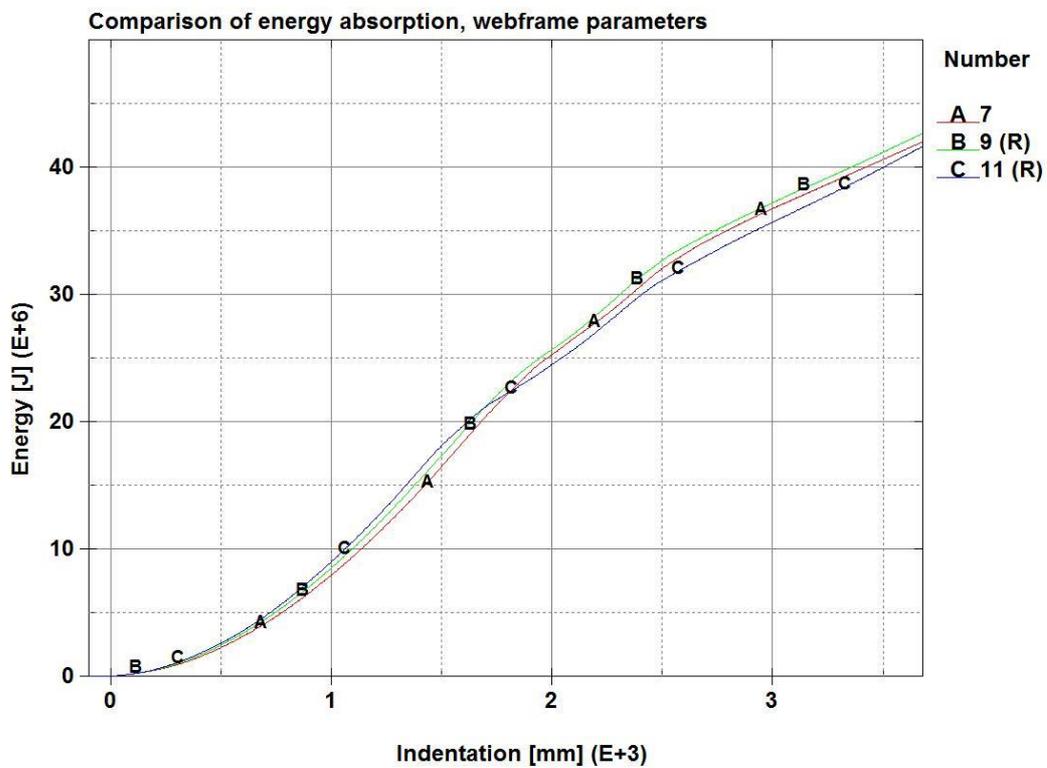
Figure 42 - Number of webframes study, modified failure strain

Thickness
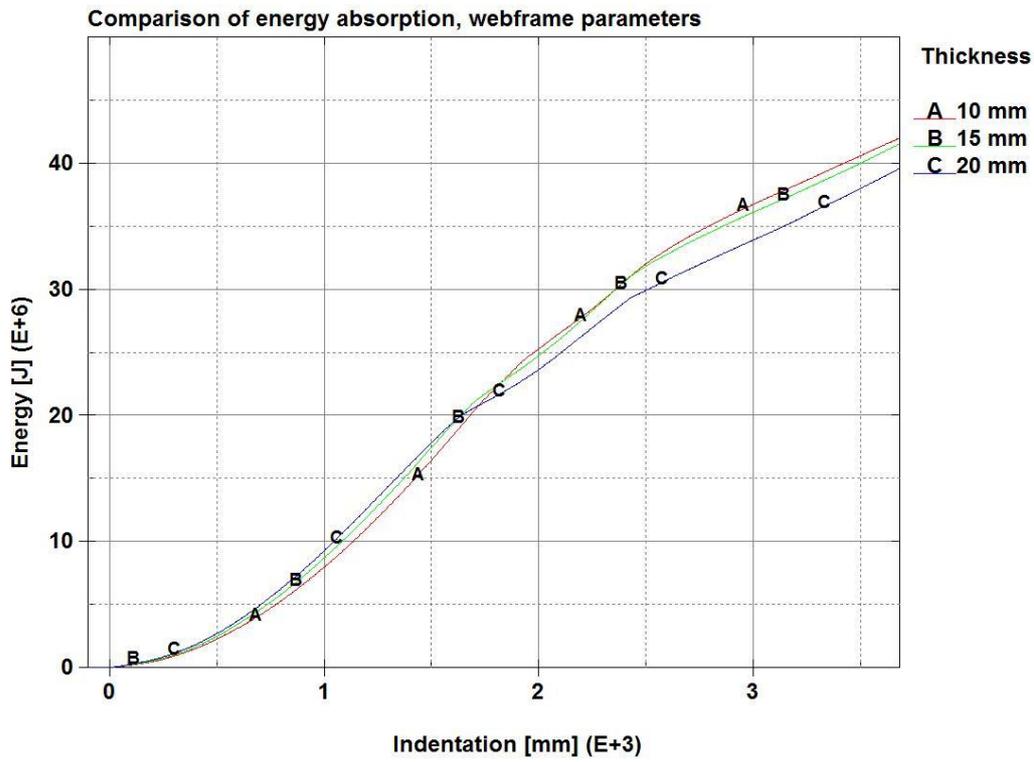
Figure 43 - Shell thickness in webframes study

Stiffener size
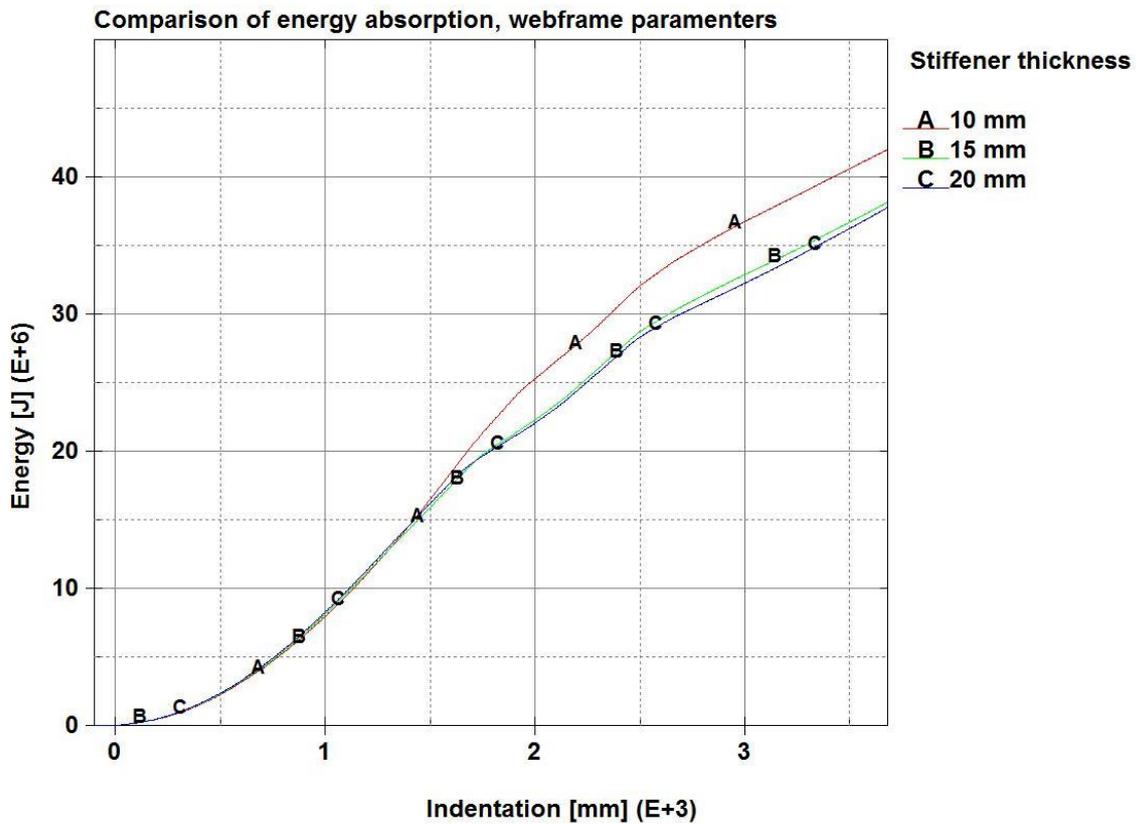
Figure 44 - Webframe stiffener study

Observations

The general trend is that increasing any single parameter in the webframes leads to a reduction of the energy absorption. The only parameter for which this does not happen is by increasing the number of webframes over the length to 9. Also here the increase is low.

7.3.2 STRINGER PARAMETERS

Number
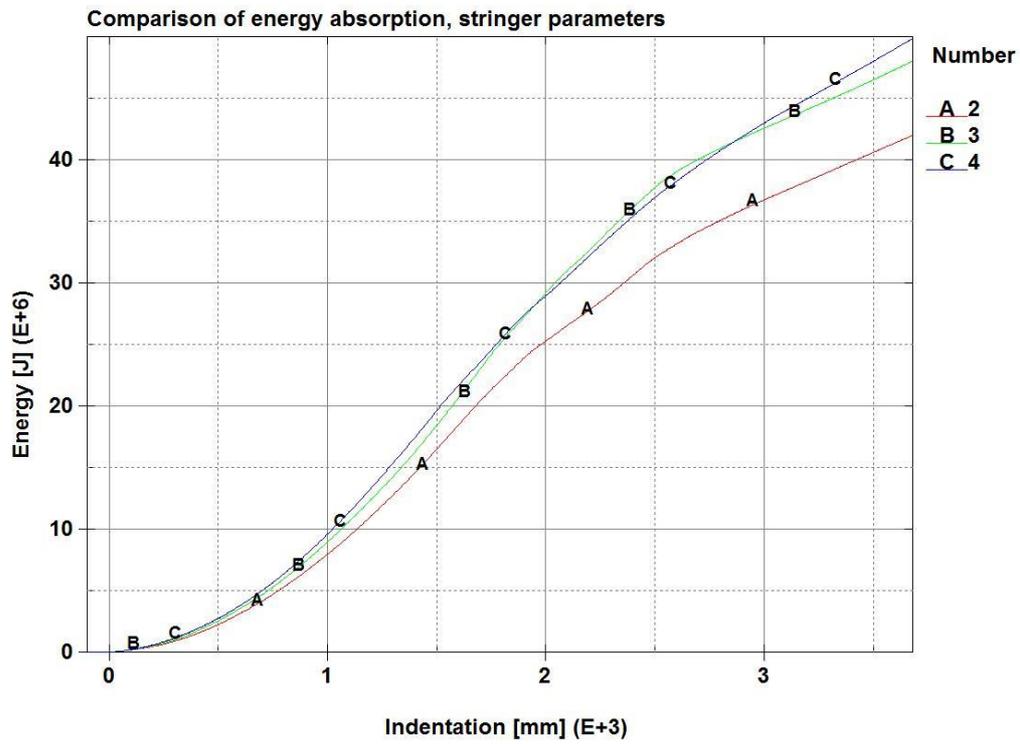
Figure 45 - Number of stringers study

Shell thickness
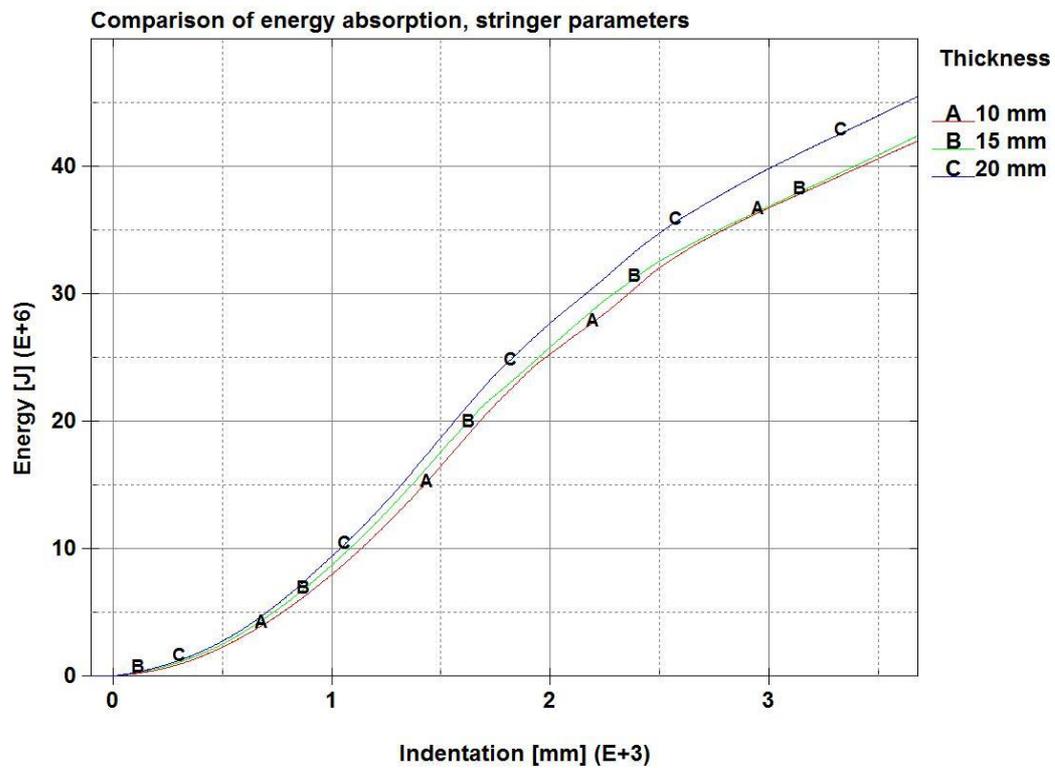
Figure 46 - Shell thickness in stringers study

Stiffener thickness

Comparison of energy absorption, stringer parameters
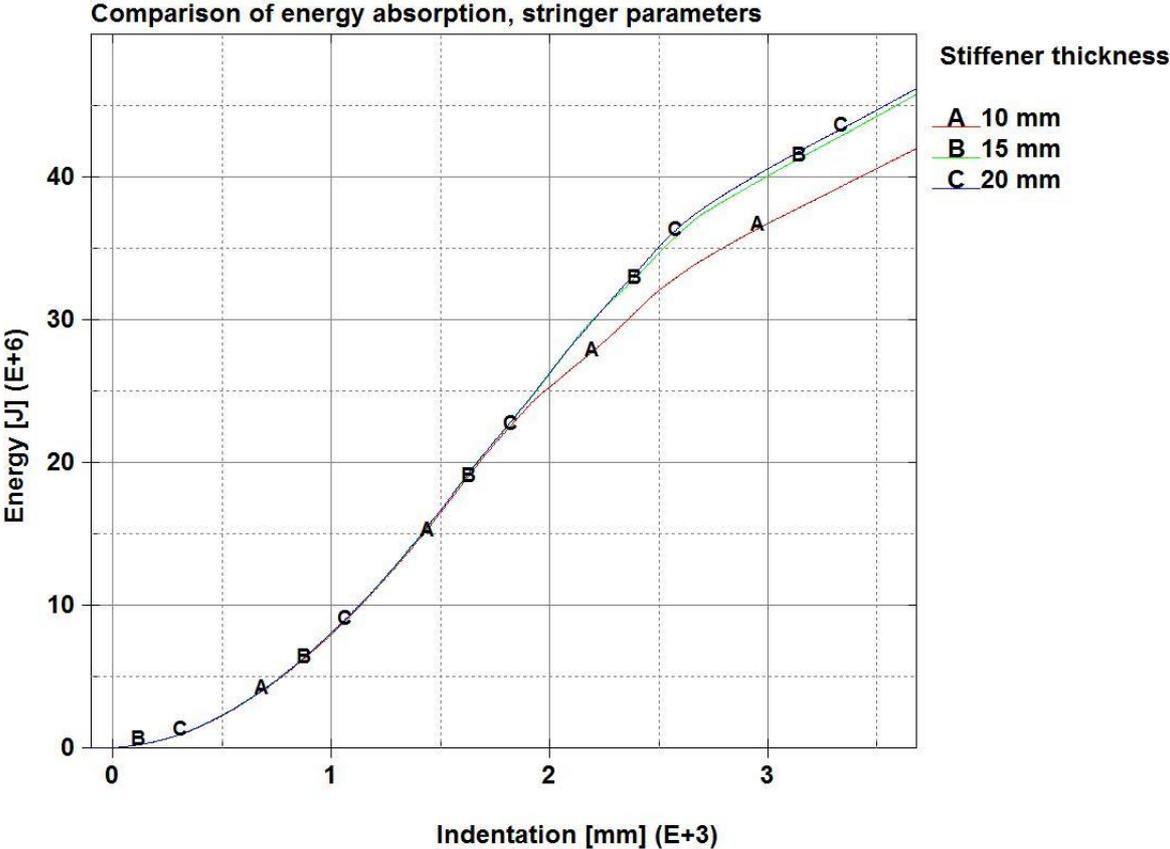
Figure 47 - Stringer stiffener study

Observations

Increasing the number of stringers seem to give significant increase from adding one stringer, the second added does not add significantly more to the energy absorption capabilities than the first.

The thickness of the skin seem to give a benefit, and in the current study it seems that the benefit from adding half of the initial thickness is trivial, while there is a significant increase by doubling it.

Adding thickness to the stiffeners seems beneficial, it is however mostly associated with the first step of the increase, and the second does not seem to add much to the energy absorption.

Summed up most of the parameters associated with the stringers are related to a better energy absorption when increased.

7.3.3 OUTER SKIN
Shell thickness
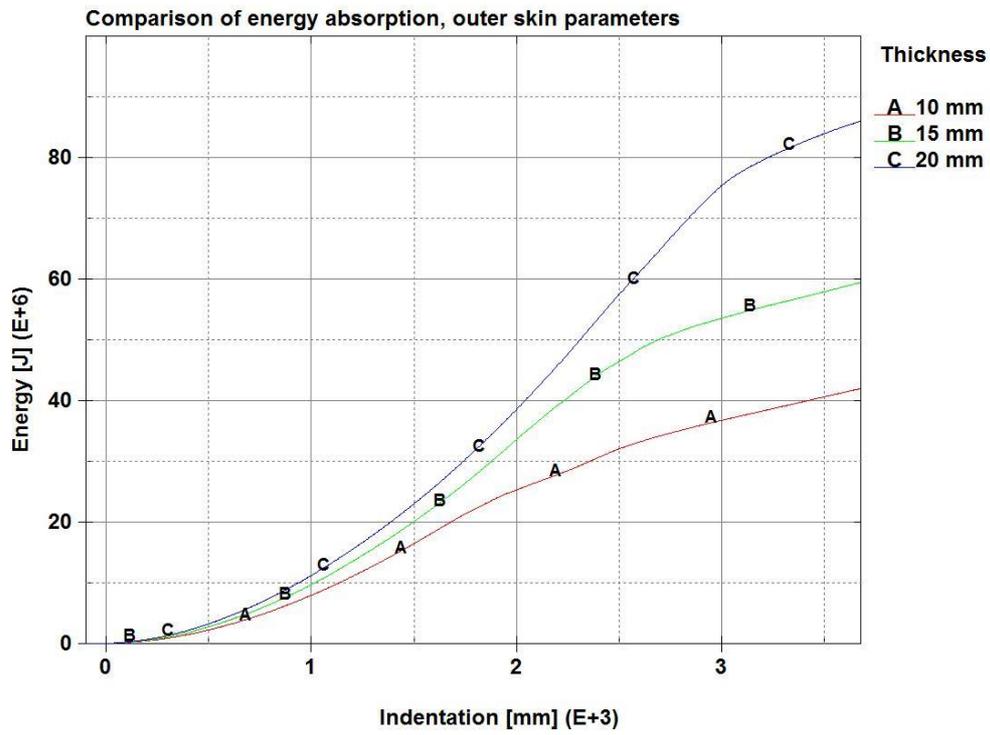
Figure 48 - Outer skin stiffener size study

Stiffener thickness
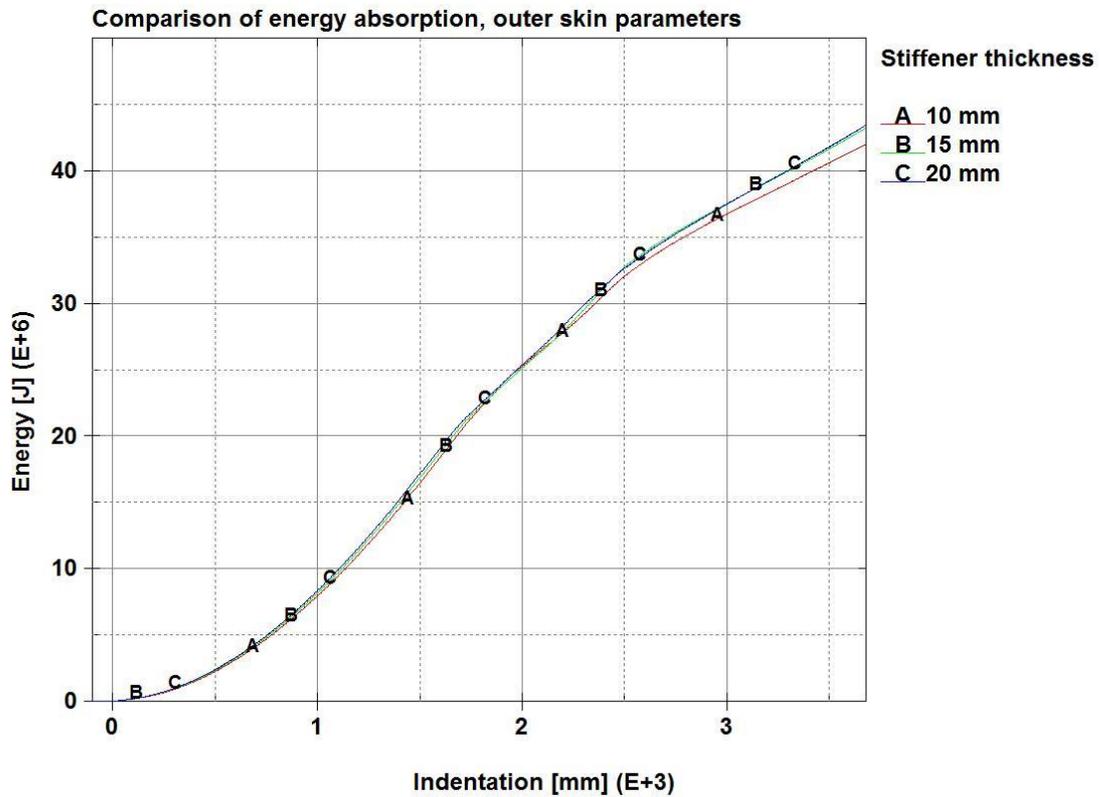
Figure 49 - Outer skin stiffener study

Number of stiffeners

Comparison of energy absorption, outer skin parameters
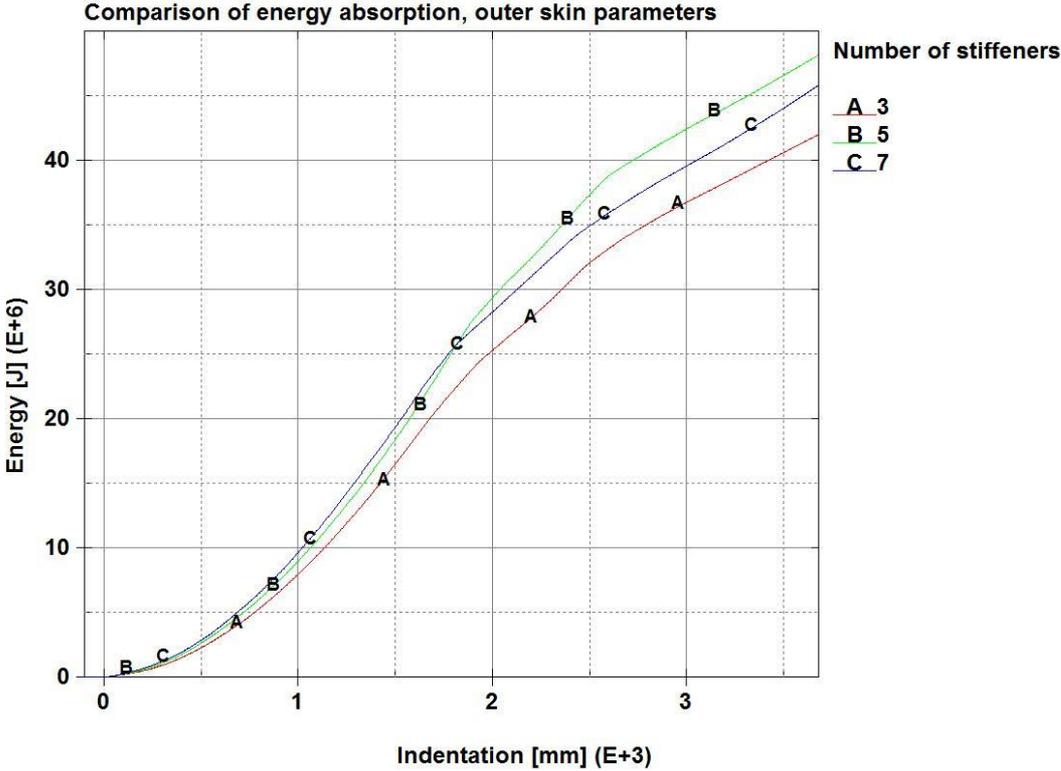
Figure 50 - Outer skin number of stiffeners study

Observations

Significant increase in crashworthiness is observed by increasing the outer skin thickness. Promising results are obtained both for the 50% increase and for the 100% increase. For the sake of this study it is noted that while significant stepwise increases in final energy level are shown, the 50% increase in plate thickness capture most of the decrease in safe distance, which is the desired outcome of the current study.

Increasing the thickness of the outer skin stiffeners show some increase in the curves. As the relative weight increase is assumed to be small and the resulting increase in energy absorption is small it is difficult to observe anything directly from the curves, other than that an increased thickness results in increased energy absorption.

Adding stiffeners seem to give better energy absorption, it is however noted that by introducing seven stiffeners over one stringer spacing the energy absorption is worse than for five.

7.3.4 INNER SKIN PARAMETERS
Shell thickness
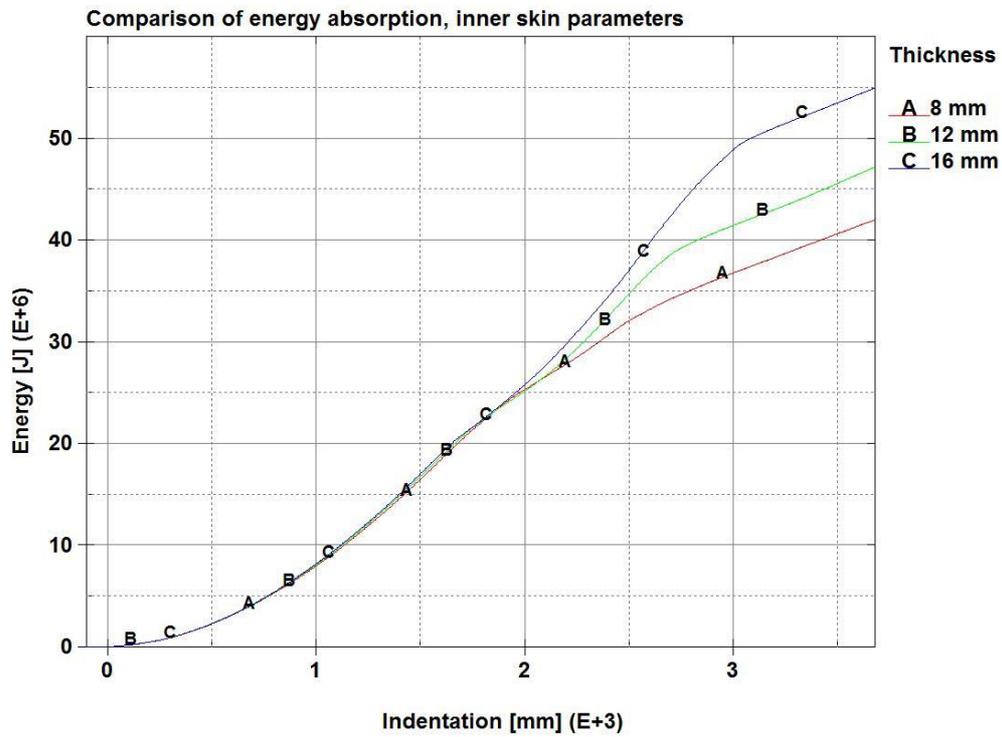
Figure 51 - Inner skin shell thickness study

Stiffener thickness
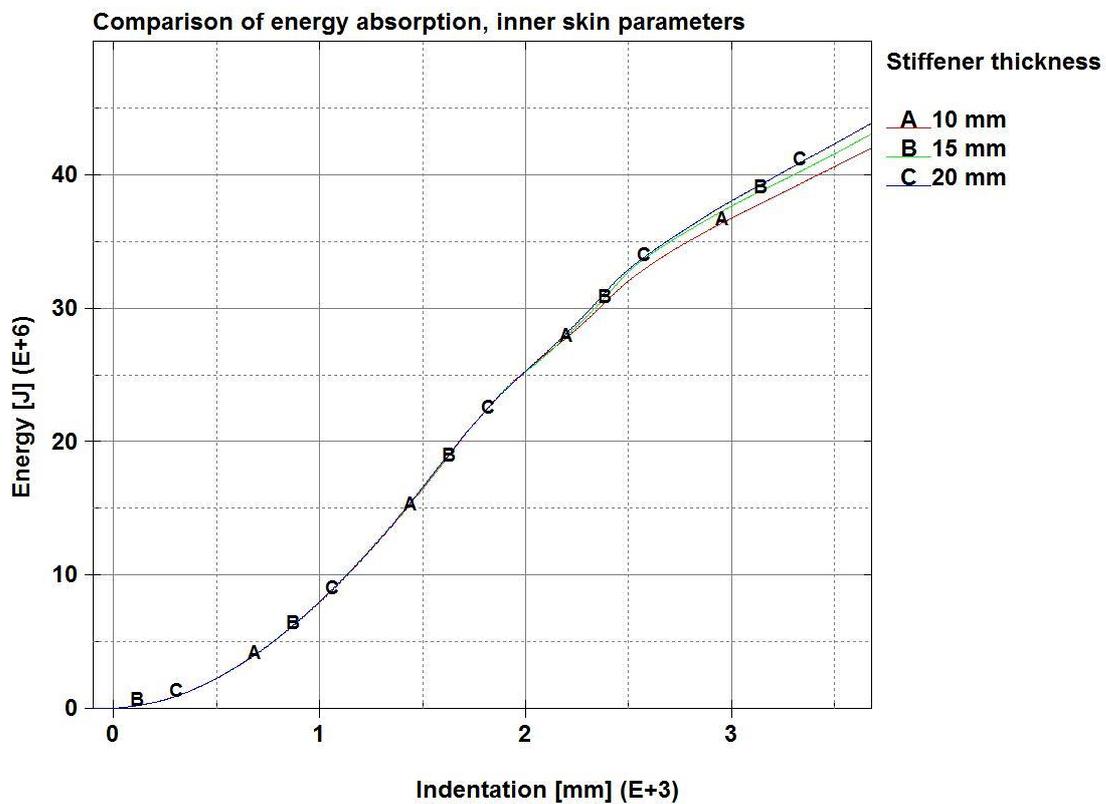
Figure 52 - Inner skin stiffener study

Number of stiffeners

Comparison of energy absorption, inner skin parameters
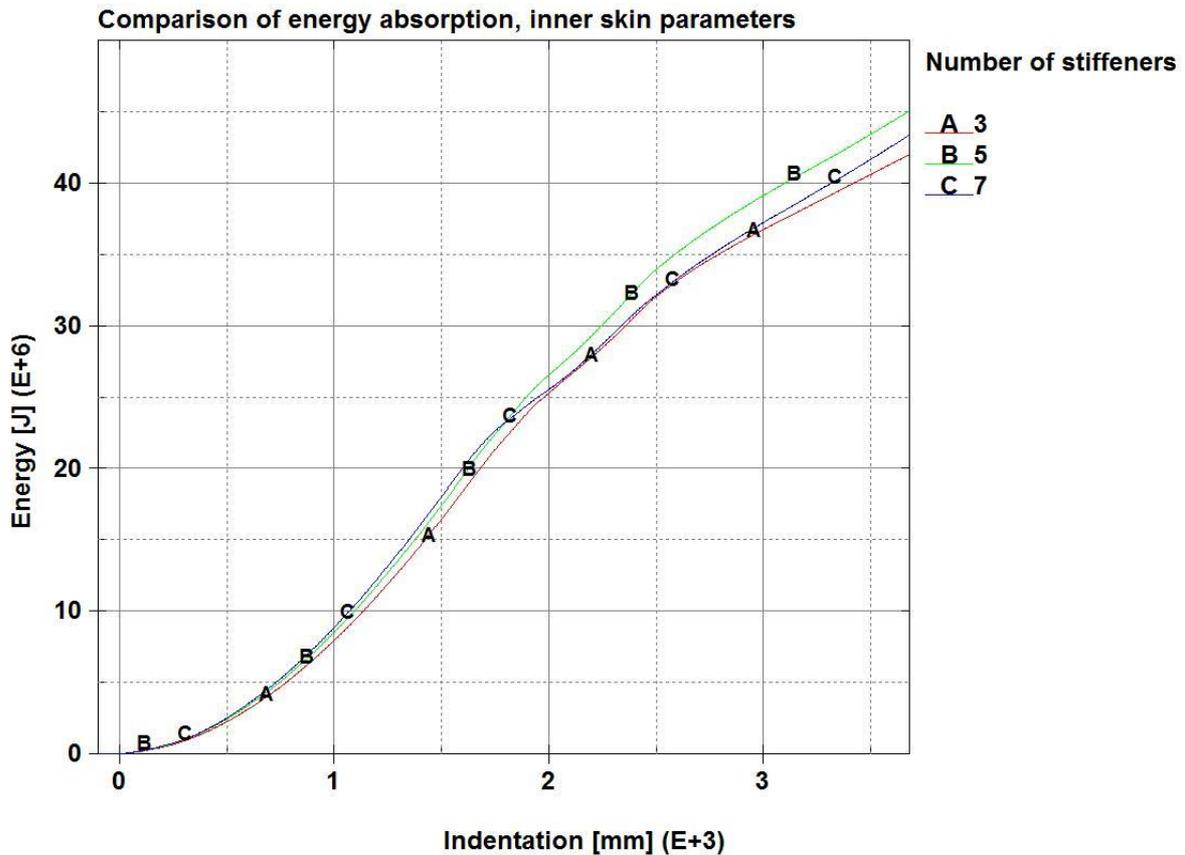
Figure 53 - Inner skin number of stiffeners study

Observations

Increasing the inner skin thickness seem to give response somewhat comparable to an increase in outer skin, however it does differ when it comes to the indentation needed for the effect to take place.

Further stiffening also follow the same pattern as the outer skin study, here as well, the indentation needed for the effect to take place is larger.

As for the other parameters additional stiffeners seem to give the same pattern as in the outer skin, only to a smaller degree.

7.4 PARAMETERS INCREASED CONCURRENTLY

Stiffeners in inner and outer skin

Increasing the number of stiffeners or thickness of these in either inner or outer skin gave odd results. A study was carried out to investigate how they behave when when they are increased concurrently. Figure 54 show the results. It is seen that the response from the first increase is significant, while the second increase in number of stiffeners only serve to increase the energy absorbed slightly.
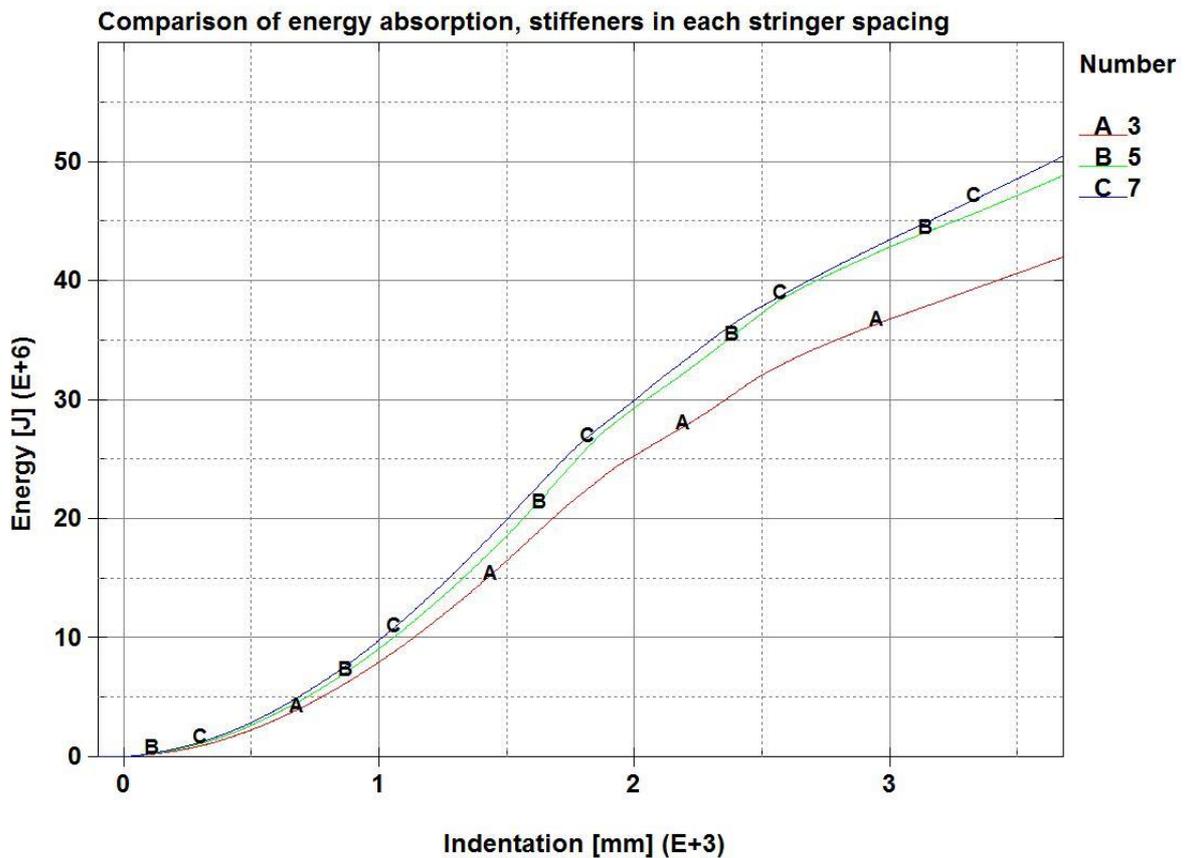
Figure 54 - Number of stiffeners study

On this matter it is concluded that the observed odd behaviour most probably arise from effects of adding stiffeners at one side. Different number of stiffeners leads to different meshes in the model, and this might explain the behaviour. Another explanation might be that earlier rupture is initialized by the structural arrangement.

Webframes and stringers

It is seen that changes in the webframe structure serves only to give a decrease, for one parameter a slight increase, of the energy absorption. This might give an indication that a stiffening of the webframe leads to earlier rupture. This should be investigated further

for verification or abandonment of this hypothesis. On the other hand; all the variables regarding stringers give significant increases, but seem to have some kind of upper bound for the increase obtained from one variable. One reason for the observed behaviour might be that a cruciform is formed by the intersections of stringers and webframes in their initial conditions. This effect might be reduced when stiffening one of the components alone, i.e. the stringers or webframes might act as girders rather than cruciform. This is easily illustrated by imagination of the stringers being made of 2mm sheet metal and the webframes consisting of 12mm steel plates, most probably the sheet metal would follow the webframes girder response, rupture and bend away. Also stress concentrations in transitions from strong to weak structural elements might give earlier rupture.

To investigate this simulations are carried out, where the webframe and stringers are modified by the same stepwise increases. One study is made; where as well as the stepwise increase of the stringer and webframe variables, the outer skin thickness is adjusted in the same steps. This is shown in figure 55 and figure 56.
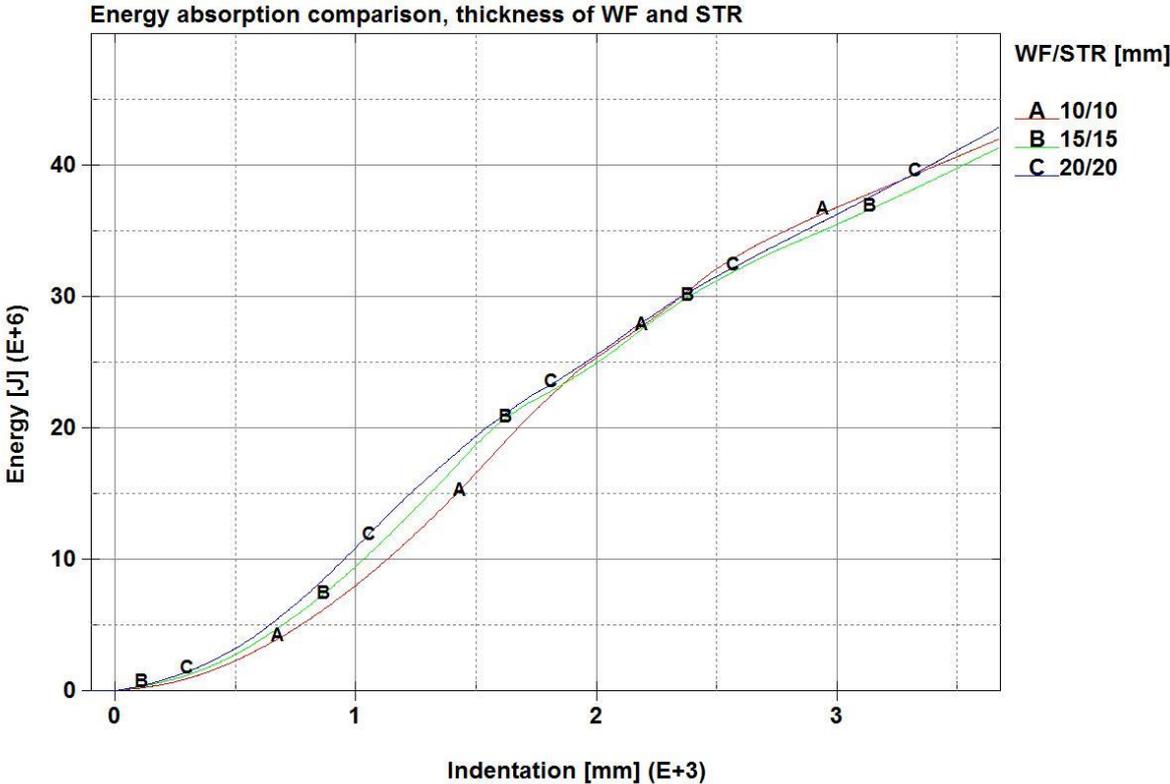
Figure 55 - Stepwise increase of webframes (WF) and stringers (STR)

From this study it is seen that the initial forces are bigger when increasing the stiffness. Rupture seem to initiate at an earlier state for increased stiffness. By also increasing the outer skin thickness, the early rupture seem to be avoided and thus significant increases are obtained. It is observed that the final amounts of energy are lower in this study than what is observed for adjusting the outer skin only. This behaviour could be studied further by accounting for stepwise increase of the inner skin as well. As the aim of this study is the reduction of the safe distance as described, this is left out of the current study.
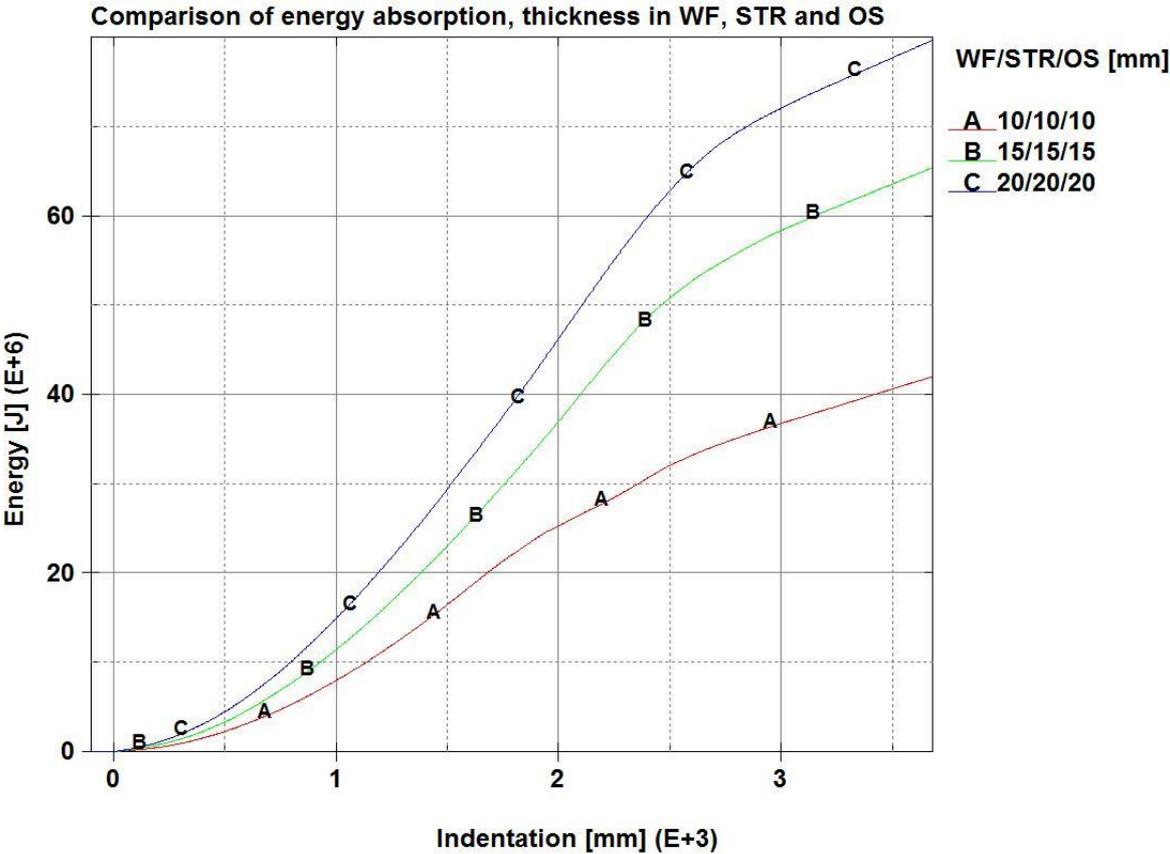
Figure 56 Stepwise increase of webframes (WF), stringers (STR) and outer skin (OS)

Weight and comparisons

Through values, read from the results files, as listed in the parameter matrix (Appendix D) the most promising parameters are identified. Table 10 presents these; it gives the different configurations, increased mass, reduction in minimum distance and a comparison ratio, namely; the reduction in minimum distance divided by the increased mass.

Results of the configurations in the Initial, Maxweight as well as the combined parameter study, including the outer skin thickness, are included.

Table 10 - Results comparison

Variable	Value	Increased mass of half section [ton]	Reduction in minimum distance [m]	Ratio [m/ton]
Initial		0	0.00	-
Maxweight		72.8	2.38	0.033
hbs	1560mm	5.1	0.79	0.156
hbs	1950mm	2.8	0.75	0.266
nssd	5	3.6	0.77	0.213
nssd	7	6.5	0.82	0.127
nssd_is	5	2.3	0.34	0.145
nssd_os	5	2.3	0.73	0.312
nssd_os	7	3.9	0.4	0.102
ssis	15mm	0.9	0.13	0.137
ssis	20mm	1.9	0.22	0.116
ssos	15mm	0.9	0.14	0.147
ssos	20mm	1.9	0.16	0.084
ssst	15mm	0.5	0.45	0.948
ssst	20mm	0.9	0.51	0.537
stis	12mm	4.1	0.61	0.148
stis	16mm	8.2	0.99	0.120
stos	15mm	5.1	1.37	0.266
stos	20mm	10.3	1.58	0.154
stst	20mm	3.5	0.32	0.092
wf/str/os	15mm	8.5	1.52	0.179
wf/str/os	20mm	17.1	1.8	0.105

7.5 Sources of errors

- The bow is modelled as a simple and rigid bulb. This might alter the structural response of the side section, and the side section accounts for the full amount of deformation energy, which is physically not necessarily correct.

- All cut-outs and manholes are modelled by means of a reduction of the plate thicknesses. This might lead to alterations in collapse pattern and altered stiffness in the areas with cut-outs.

- The bulb stiffener profiles modelled as flat bar stiffeners with the equivalent area, giving a slightly lower resistance against bending.

- When changing the geometry re-meshing is carried out. When the quality of the new mesh differs from the initial mesh, differences in the results might arise.

- Human error. The simulations and post processing include interaction between four different programs. Automatic model checking by means of LS-PREPOST has been used for verification. But still, the complete simulation setup remains complex.

8 DISCUSSION

Studies are available from the literature studying the quantification and maximization of the energy absorbed by a side structure with different failure criteria, for example the rupture of the inner skin. The aim of the current study is to identify the most effective parameters to vary when the reduction of the safe distance of sensitive equipment is desired. It is seen that with the boundaries set there exist large potential for increasing the crashworthiness. Adjusting the failure strain modifies the result only slightly, and considering the error sources given in section 7.5 the resulting graphs are given confidence for comparison studies.

From table 10 it is seen that as a standalone parameter, the outer skin thickness could be increased, and gives a significant decrease of the safe distance. On this matter it should be noted that the outer shell thickness is studied as constant over the height of the side. Possibly it is not necessary to implement the increase over the whole height. A variation could give a better decreased indentation to added weight ratio and should be studied closer. Another standalone parameter which show promising results is the introduction of an extra stringer.

Stiffening the stringers also seems a good option, looking at the high comparison ratio. The mechanism accountable for this might be the crushing process of the web girders created by the stringers. Effect of longitudinal stiffeners in web girders is discussed by Hong and Amdahl, they conclude from a numerical study of stiffened web girders that *"closely spaced stiffeners will disturb the crushing process to some extent" (50)*. It should however be noted that in the current study these are somewhat crudely modelled due to the cut-out modelling, and need closer investigation. Such behaviour is assumed to be highly case dependent, and might be unpractical for engineering applications, as the needed modelling for each case is substantial. It is for the remainder of this thesis this variable is left as promising, but to unsure for conclusion. In this case the indication is that almost all increase in structural parameters regarding the stringers are good options.

From the analytical formulae, as discussed in section 5.1, it can be seen that in the context of simplified methods the resistance force from cruciform and stringers are independent of the indentation. The same for the outer skin is dependent on this. To

obtain a lower indentation for the absorption of collision energy, high force from the start of the collision is important, as the energy is the force integrated over distance. Logically this leads to the hypothesis that the strengthening of the stringers and webframes concurrently is beneficial for the current aim. The numerical studies, however, reject this for the current case. It should be noted that the consideration of deformation of the bow might change this, as at some point the structure would comply with the strength design requirements, and the bow would account for more of the dissipated energy.

Increasing the outer skin thickness as well as the thickness in webframes and stringers, seem to give good results. It is noted that in terms of final energy absorption this gives a lower measure than by increasing only the outer skin thickness, still the reduction in safe distance is better. This could be explained by the fact that the inner skin is left unchanged. It could be studied further, but does not benefit the aim of this thesis and therefore left out.

Implementation of special core structures also needs to be addressed. This has not been studied numerically in current thesis, and at this point only a discussion is attempted. In their study (6) Hogström et al. presents energy indentation curves as cited in figure 21. Here it can be observed that the energy of final indentation for the reference structure is reached for the X-core structure at about 0.37m while full indentation of the reference structure is about 0.48m. This corresponds to a reduction of 0.11m, 23% of full indentation. From Ehlers et al. (7) it is observed from the curves presented that in the tanker case, it does not seem conclusive that the novel structure would benefit the goal of the current thesis. Based on what has been assessed, it is not possible to conclude if the implementation of core structures would benefit the aim of the current thesis. It is however noted that it appears to be case dependent, and should be further studied.

9 CONCLUSION

Modelling and analysis of right angled ship collisions with a rigid bulb have been carried out. Modelling and simulation tools used, i.e. MATLAB, PATRAN, LS-DYNA and LS-PREPOST worked well together, and proved valuable for this kind of study. Convergence testing and verification by means of analytical methods show that the simulations have a sufficient degree of accuracy for comparison studies.

By parameter study, case study of ice-strengthened design and evaluation of previous studies regarding special structural elements the following measures are identified as valuable options for reducing the safe distance:

- Increase of the outer skin thickness.
- Introduction of an extra stringer.
- Increase the thicknesses in outer skin, stringers and webframes concurrently.
- Implementation of ice class or ice stiffening.

The numerical studies also indicate that parameters should be increased concurrently. The reason for this is not fully assessed, but might be due to earlier fracture when increasing single parameters.

The following points are outlined for further work:

- Closer investigation of stiffeners attached to the stringers.
- Accounting for deformations in the colliding bow structure.
- Effect of varying the thickness in outer and inner skin as well as the stringers and webframes.
- Application of different thicknesses, stiffener sizes and geometrical data over the height of the section.
- Implementing high strength steel in parts of the structure.
- Implementation of a core structure in the current model.
- Fitting a more sophisticated fracture model to the simulations.
- Using an optimization scheme with the maximization of the comparison ratio as goal function.

10 REFERENCES

1.   Mathworks.   Matlab   2013.   Available   from: http://www.mathworks.se/products/matlab/.
2.   MSCsoftware.   Patran   2012.   Available   from: http://www.mscsoftware.com/Products/CAE-Tools/Patran.aspx.
3.   LSTC I. LS-DYNA 2012. Available from: http://www.lstc.com/products/ls-dyna.
4.   Livermore STC. LS-PrePost | Online Documentation 2012. Available from: http://www.lstc.com/lspp.
5.   IUMI. Casualty and World Fleet

Statistics   as   at   01.01.2012   2012.   Available   from: http://www.iumi.com/images/stories/IUMI/Pictures/Committees/FactsFigures/Statistics/2012/SpringEdition/iumi%20casualty%20and%20world%20fleet%20statistics_spring2012.pdf.
6.   Hogström P, Ringsberg JW. Assessment of the crashworthiness of a selection of innovative ship structures. Ocean Engineering. 2013 //;59:58-72.
7.   Ehlers S, Tabri K, Schillo N, Ranta J. Implementation of a novel ship side structure into a tanker and a ropax vessel for increased chreashworthy. 6th LS-DYNA Users' Conference2007.
8.   Ehlers S. A procedure to optimize ship side structures for crashworthiness. Proceedings of the Institution of Mechanical Engineers Part M: Journal of Engineering for the Maritime Environment. 2010 //;224(1):1-11.
9.   Kitamura O. Comparative study on collision resistance of side structure. Marine Technology. 1997 //;34(4):293-308.
10.   DNV. Rules for classification of Ships. Gas Fuelled Ship Installations2012.
11.   GL. Rules for Classification and Construction, VI Additional Rules and Guidelines. Machinery Installations, Guidelines for the Use of Gas as Fuel for Ships2010.
12.   Minorsky VU. AN ANALYSIS OF SHIP COLLISIONS WITH REFERENCE TO PROTECTION OF NUCLEAR POWER PLANTS. 1959 NP-7475 Country unknown/Code not availableTue Feb 05 17:15:27 EST 2008DTIE; NSA-13-013093English.
13.   Petersen MJ. Dynamics of ship collisions. Ocean Engineering. 1982 //;9(4):295-329.
14.   Tabri K. Influence of coupling in the prediction of ship collision damage. Ships and Offshore Structures. 2012 //;7(1):47-54.
15.   Pedersen PT, Zhang S. On impact mechanics in ship collisions. Marine Structures. 1998 //;11(10):429-49.
16.   Hong L, Amdahl J, Wang G. A direct design procedure for FPSO side structures against large impact loads. Journal of Offshore Mechanics and Arctic Engineering. 2009 //;131(3):1-12.
17.   NORSOK. N-004 Design of steel structures. 2004.
18.   Pill I, Tabri K. Finite element simulations of ship collisions: A coupled approach to external dynamics and inner mechanics. Ships and Offshore Structures. 2011 //;6(1-2):59-66.
19.   Tabri K, Määttänen J, Ranta J. Model-scale experiments of symmetric ship collisions. Journal of Marine Science and Technology. 2008 //;13(1):71-84.
20.   Ehlers S. A particle swarm algorithm-based optimization for high-strength steel structures. Journal of Ship Production. 2012 //;28(1):1-9.

21.   Kennedy J, Eberhart R, editors. Particle swarm optimization1995; Perth, Aust: IEEE.

22.   Eberhart R, Kennedy J, editors. New optimizer using particle swarm theory1995; Nagoya, Jpn: IEEE.

23.   DNV. Rules for Classification of Ships 2013 [cited 2013]. Available from: http://exchange.dnv.com/publishing/rulesship/.

24.   GL. Rules for Classification and Construction Ship Technology, Ship Technology. Seagoing Ships, Hull Structures2012.

25.   Zhang L, Egge E, D., Bruhns H. Approval Procedure Concept for Alternative Arrangements. ICCGS; Tokyo, Japan2004.

26.   Janse SU. Procedure for the lateral arrangement of gas fuel tanks with the background of collision safety. 2012.

27.   DNV. Strength analysis of hull structures in tankers, Classification notes no. 31.3. 1999.

28.   DNV. OFFSHORE STANDARD DNV-OS-A101. SAFETY PRINCIPLES AND ARRANGEMENTS2011.

29.   GL. Rules for classification and construction, Industrial Services. Offshore Technology, Structural Design2007.

30.   Lloyd'sRegister. Future IMO legislation. Lloyd's Register, 2012.

31.   Wriggers P. Nichtlineare Finite-Element-Methoden: Springer; 2001.

32.   Moan T. Finite element modelling and analysis of marine structures. Trondheim: Marinteknisk senter.; 2003. 1 b. (flere pag.) : ill. p.

33.   Saabye Ottosen N, Petersson H. Introduction to the finite element method. Harlow: Pearson Prentice Hall; 1992. XV, 410 s. : ill. p.

34.   Haris S, Amdahl J. Analysis of ship-ship collision damage accounting for bow and side deformation interaction. Marine Structures. 2013 //;32:18-48.

35.   LSTC I, Hallquist J, O. LS-DYNA theory manual: Livermore Software Technology Corporation; 2006.

36.   Hughes TJR, Cohen M, Haroun M. Reduced and selective integration techniques in the finite element analysis of plates. Nuclear Engineering and Design. 1978 //;46(1):203-22.

37.   Belytschko T, Lin JI, Chen-Shyh T. Explicit algorithms for the nonlinear dynamics of shells. Computer Methods in Applied Mechanics and Engineering. 1984 //;42(2):225-51.

38.   Flanagan DP, Belytschko T. UNIFORM STRAIN HEXAHEDRON AND QUADRILATERAL WITH ORTHOGONAL HOURGLASS CONTROL. International Journal for Numerical Methods in Engineering. 1981 //;17(5):679-706.

39.   Hughes TJR, Liu WK. Nonlinear finite element analysis of shells: Part I. three-dimensional shells. Computer Methods in Applied Mechanics and Engineering. 1981 //;26(3):331-62.

40.   Hughes TJR, Liu WK. Nonlinear finite element analysis of shells-part II. two-dimensional shells. Computer Methods in Applied Mechanics and Engineering. 1981 //;27(2):167-81.

41.   Alsos HS, Amdahl J, Hopperstad OS. On the resistance to penetration of stiffened plates, Part II: Numerical analysis. International Journal of Impact Engineering. 2009 //;36(7):875-87.

42.   Ehlers S, Broekhuijsen J, Alsos HS, Biehl F, Tabri K. Simulating the collision response of ship side structures: A failure criteria benchmark study. International Shipbuilding Progress. 2008 //;55(1-2):127-44.

43.   Ohtsubo H, Kawamoto Y, Kuroiwa T. Experimental and numerical research on ship collision and grounding of oil tankers. Nuclear Engineering and Design. 1994 //;150(2-3):385-96.
44.   Zheng Y, Aksu S, Vassalos D, Tuzcu C. Study on side structure resistance to ship-ship collisions. Ships and Offshore Structures. 2007 //;2(3):273-93.
45.   Wu F, Spong R, Wang G. Using Numerical Simulation to Analyze Ship Collisions. Proceedings of the 3rd International Conference on Collision and Grounding of Ships : ICCGS 2004; 25-27 october 2004; Izu, Japan: SNAME, Japan; 2004. p. 27-33.
46.   Livermore STC. LS-DYNA keyword user's manual2007.
47.   Haris S, Amdahl J. An analytical model to assess a ship side during a collision. Ships and Offshore Structures. 2012 //;7(4):431-48.
48.   Konter A, Broekhuijsen J, Vredeveldt A. A quantitative assessment of the factors contributing to the accuracy of ship collision predictions with the finite element method

Proceedings of the 3rd International Conference on Collision and Grounding of Ships : ICCGS 2004: SNAME, Japan; 2004. 327 s. : ill. p.
49.   Wang G, Arita K, Liu D. Behavior of a double hull in a variety of stranding or collision scenarios. Marine Structures. 2000 //;13(3):147-87.
50.   Hong L, Amdahl J. Crushing resistance of web girders in ship collision and grounding. Marine Structures. 2008 //;21(4):374-401.
51.   Zhang S. The mechanics of ship collisions: Technical University of Denmark; 1999.
52.   Haris S, Amdahl J. Crushing resistance of a cruciform and its application to ship collision and grounding. Ships and Offshore Structures. 2012 //;7(2):185-95.
53.   Hayduk RJ, Wierzbicki T. EXTENSIONAL COLLAPSE MODES OF STRUCTURAL MEMBERS. Computers and Structures. 1984;18(3):447-58.
54.   Karlsson UB, Ringsberg JW, Johnson E, Hoseini M, Ulfvarson A. Experimental and numerical investigation of bulb impact with a ship side-shell structure. Marine Technology. 2009 //;46(1):16-26.
55.   Klanac A, Ehlers S, Tabri K, Rudan S, Broekhuijsen J, editors. Qualitative design assessment of crashworthy structures2005; Lisboa.
56.   LSTC I, DYNAmore G. LS-DYNA Examples 2012. Available from: http://www.dynaexamples.com/.
57.   Paik JK. Practical techniques for finite element modeling to simulate structural crashworthiness in ship collisions and grounding (Part I: Theory). Ships and Offshore Structures. 2007 //;2(1):69-80.
58.   Klanac A, Ehlers S, Jelovica J. Optimization of crashworthy marine structures. Marine Structures. 2009 //;22(4):670-90.
59.   Hogström P, Ringsberg JW, Johnson E. Survivability analysis of a struck ship with damage opening - influence from model and material properties uncertainties. Ships and Offshore Structures. 2011 //;6(4):339-54.
60.   DNV. Rules for classification of Ships. Ships for Navigation in Ice2012.

11.1 LIST OF FIGURES

11.2 LIST OF FORMULAS

11.3 LIST OF TABLES

12 Appendix

A – Drawings

Typical Midship Section

PRINCIPAL PARTICULARS

LENGTH OVER ALL .... 103.80 m
LENGTH BETWEEN PERPENDICULARS .... 101.60 m
BREADTH MLD. .... 18.40 m
DEPTH MLD. MAIN DECK .... 9.05 m
DRAUGHT, DESIGN .... 5.10 m
DRAUGHT, SCANTLING .... 5.30 m
MAXIMUM SERVICE SPEED .... 12.5 knots
BLOCK COEFFICIENT .... 0.75 _

STEEL QUALITY:
IN GENERAL, MILD STEEL NV-NS GRADE A UNO.

TYPICAL WEB FRAME
MAX WEB FRAME SPACING 3100MM

TYPICAL ORDINARY FRAME
FRAME SPACING 700MM

TANK TOP

STRINGER 1
(STRINGER 2 SIMILAR)
PL 14.0 MTRL/HD

TYP. LONG GIRDER IN DB
PL 12.0 MTRL/HD

TYP. LONG SECTION IN DB

STEEL QUALITY:
IN GENERAL, MILD STEEL IN-NS GRADE A, UNO.

PRINCIPAL PARTICULARS

LENGTH, OVER ALL	103.80	m
LENGTH, BETWEEN PERPENDICULARS	101.60	m
BREADTH, MLD.	18.40	m
DEPTH MLD, MAIN DECK	9.05	m
DRAUGHT, DESIGN	5.10	m
DRAUGHT, SCANTLING	5.30	m
MAXIMUM SERVICE SPEED	12.5	knots
BLOCK COEFFICIENT	0.75	–
ICE CLASS	ICE–1A	

Rolls-Royce

Typical Midship Section
Longitudinal Ice Frames

SCALE 1:50
A1

PAGE: 1/1   REV: 0

TANK TOP

STRINGER 1
(STRINGER 2 SIMILAR)
PL 11,5/FB100x10

TYP. LONG. SECTION IN DB

TYP. LONG. GIRDER IN DB
PL 11,5/FB100x10

TYPICAL WEB FRAME
WEB FRAME SPACING 2100MM

TYPICAL ORDINARY FRAME
FRAME SPACING 700MM

STEEL QUALITY:
IN GENERAL, MILD STEEL NV-NS GRADE A UNO.

PRINCIPAL PARTICULARS

LENGTH OVER ALL	103,80 m
LENGTH BETWEEN PERPENDICULARS	101,60 m
BREADTH MLD.	18,40 m
DEPTH MLD. MAIN DECK	9,05 m
DRAUGHT, DESIGN	5,10 m
DRAUGHT, SCANTLING	5,30 m
MAXIMUM SERVICE SPEED	12,5 knots
BLOCK COEFFICIENT	0,75 —
ICE CLASS	ICE-1A

B – ZIP FILE

As the result files from one collision simulation are bigger than the allowed appendix file size, result files are not provided.

This appendix consists of a .zip file containing the following:

1)   Drawings from appendix A.

2)   Matlab code and input files.

3)   Example folder for a simulation in LS-DYNA, P1 initial condition. Animation and database files are removed, due to their size. The in LS-DYNA executable input files are "finishedfile.key" in each bow position folder.

4)   Poster.

C – Force indentation curves for each bow position in the initial configuration

Force indentation plot, bow position 1, P1

Force indentation plot, bow position 1, P2
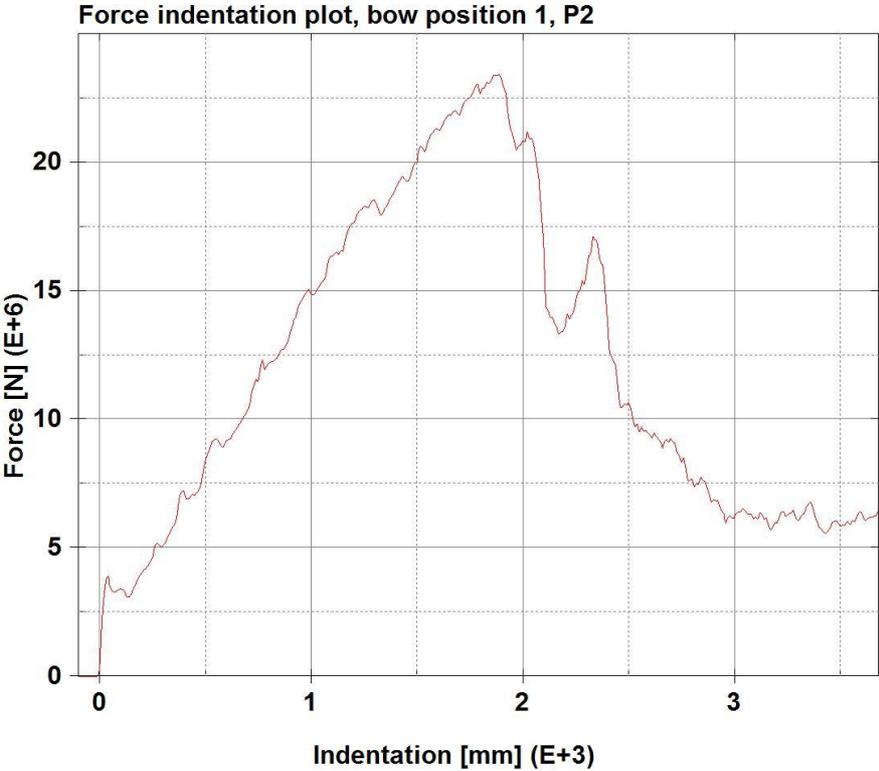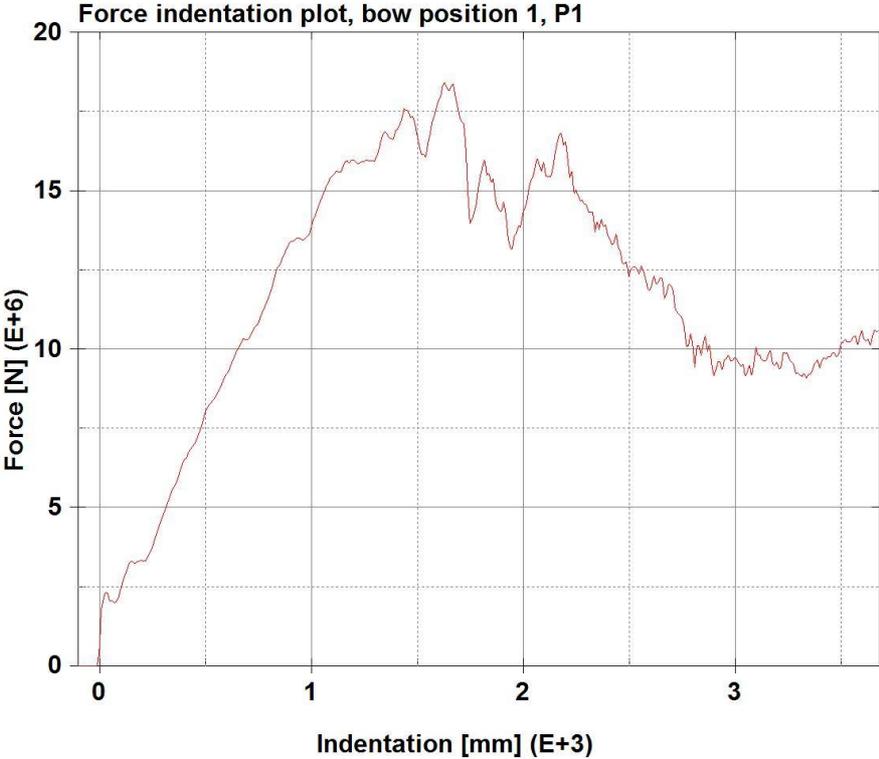
Force indentation plot, bow position 3, P1
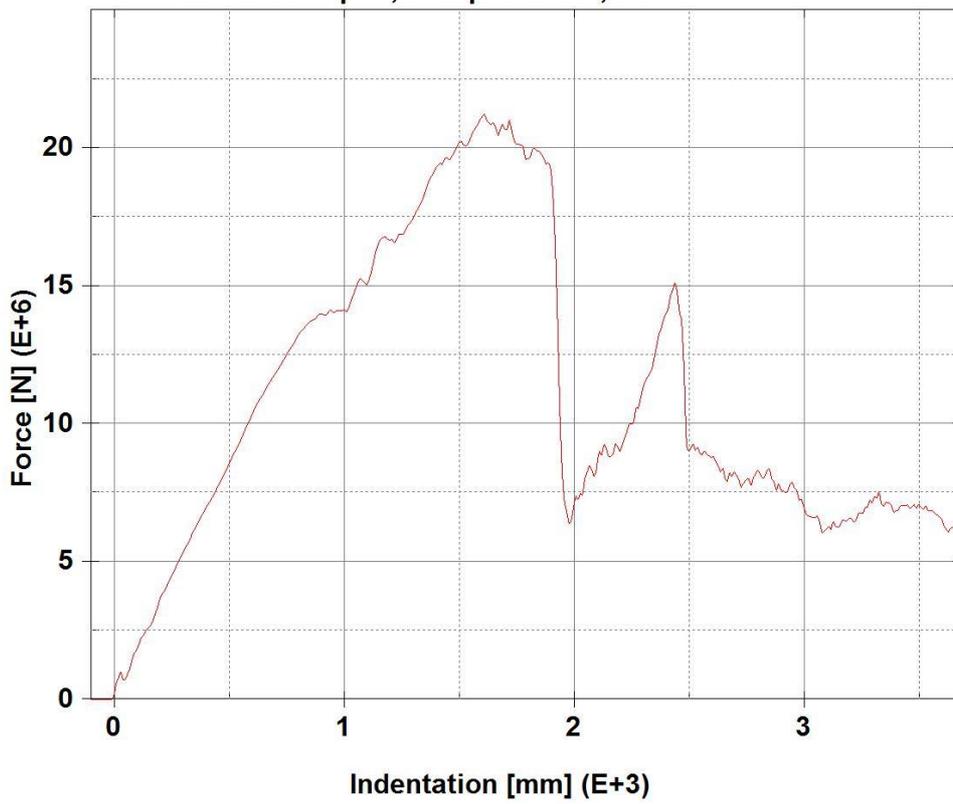
Force indentation plot, bow position 4, P1
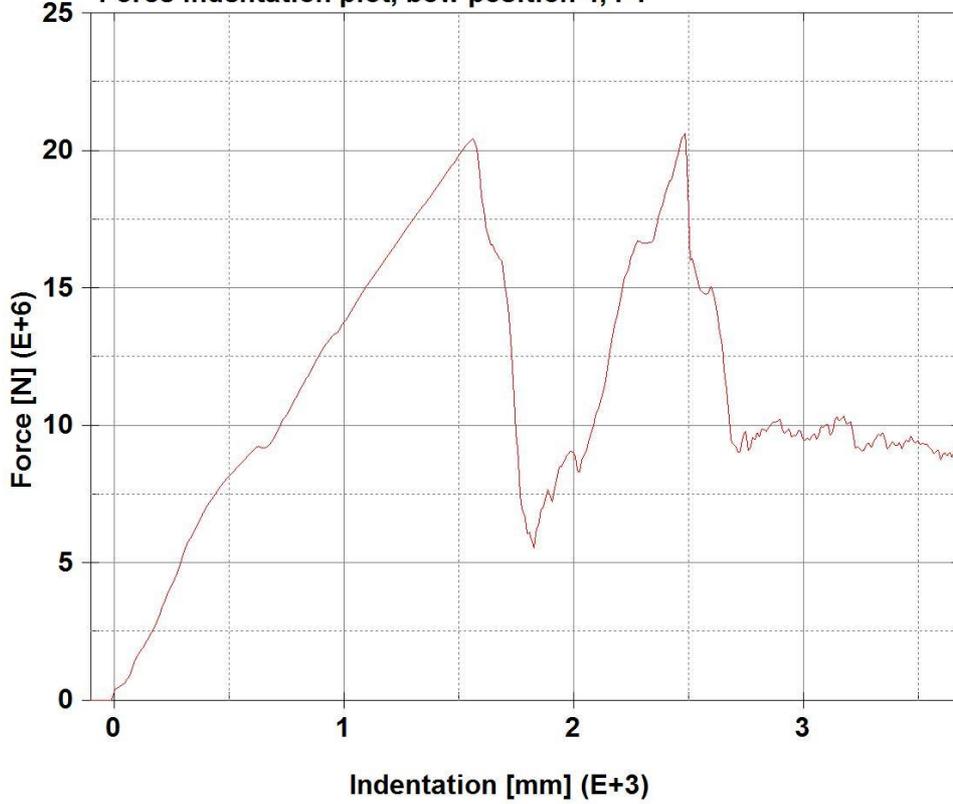
D – PARAMETER MATRIX

Parameter matrix setup

Variables	Number				
	Abbreviation	Initial configuration	min	max	dimension
Length between webframes	lbwf	2100,00	1400,00	2100,00	mm
Height between stringers	hbs	2600,00	1560,00	2600,00	mm
Number of stiffeners in each stringer distance outer skin	nssd_os	3,00	3,00	7,00	-
Number of stiffeners in each stringer distance inner skin	nssd_is	3,00	3,00	7,00	-
Shell thickness of outer skin	stos	10,00	10,00	20,00	mm
Shell thickness of inner skin	stis	8,00	8,00	16,00	mm
Shell thickness of web frame	stwf	10,00	10,00	20,00	mm
Shell thickness of stringers	stst	10,00	10,00	20,00	mm
Stiffener thickness in outer skin	ssos	10,00	10,00	20,00	mm
Stiffener thickness in inner skin	ssis	10,00	10,00	20,00	mm
Stiffener thickness in web frame	sswf	10,00	10,00	20,00	mm
Stiffener thickness in stringers	ssst	10,00	10,00	20,00	mm

Parameter matrix setup

Variables	1,00	2,00	2R	3,00	3R	4,00	4R	5,00	6,00	6R	7,00
Length between webframes	2100,00	1400,00	1400,00	1680,00	1680,00	1400,00	1400,00	2100,00	2100,00	2100,00	2100,00
Height between stringers	2600,00	1560,00	1560,00	2600,00	2600,00	2600,00	2600,00	1950,00	1560,00	1560,00	2600,00
Number of stiffeners in each stringer distance outer skin	3,00	7,00	7,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	5,00
Number of stiffeners in each stringer distance inner skin	3,00	7,00	7,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	5,00
Shell thickness of outer skin	10,00	20,00	20,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00
Shell thickness of inner skin	8,00	16,00	16,00	8,00	8,00	8,00	8,00	8,00	8,00	8,00	8,00
Shell thickness of web frame	10,00	20,00	20,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00
Shell thickness of stringers	10,00	20,00	20,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00
Stiffener thickness in outer skin	10,00	20,00	20,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00
Stiffener thickness in inner skin	10,00	20,00	20,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00
Stiffener thickness in web frame	10,00	20,00	20,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00
Stiffener thickness in stringers	10,00	20,00	20,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00
Built	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Run	Yes	Yes		Yes		Yes		Yes	Yes		Yes
Postprocessing	Yes	Yes		Yes		Yes		Yes	Yes		Yes
Comment											
Smallest element edge size	80,00	75,00	75,00	73.3 (57)	73.3 (57)	75,00	75,00	80,00	78,00	78,00	80,00
Fs	0.29	0.29	0.295	0.29	0.296	0.29	0.295	0.29	0.29	0.292	0.29
Weight	94,128	166,94	166,943	98,3115	98,312	102,49	102,49	96,95	99,1885	99,189	97,735
Increase	0	72,816	72,8155	4,184	4,184	8,3655	8,3655	2,8223	5,061	5,061	3,607
Energy absorbed at max indentation	42,04	120,6	121,94	41,89	42,72	40,12	41,69	48,09	49,92	50,12	48,9
Safe distance	3,68	1,3	1,3		3,58			2,93	2,91	2,89	2,91
Reduction in safe distance	0,00	2,38	2,38		0,10			0,75	0,77	0,79	0,77
Comparison ratio	-	0,0327	0,032685		0,0239			0,2657	0,15214	0,1561	0,2135

Parameter matrix setup

Variables	8,00	9,00	10,00	11,00	12,00	13,00	14,00	15,00	16,00	17,00	18,00
Length between webframes	2100,00	2100,00	2100,00	2100,00	2100,00	2100,00	2100,00	2100,00	2100,00	2100,00	2100,00
Height between stringers	2600,00	2600,00	2600,00	2600,00	2600,00	2600,00	2600,00	2600,00	2600,00	2600,00	2600,00
Number of stiffeners in each stringer distance outer skin	7,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00
Number of stiffeners in each stringer distance inner skin	7,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00
Shell thickness of outer skin	10,00	15,00	20,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00
Shell thickness of inner skin	8,00	8,00	8,00	12,00	16,00	8,00	8,00	8,00	8,00	8,00	8,00
Shell thickness of web frame	10,00	10,00	10,00	10,00	10,00	15,00	20,00	10,00	10,00	10,00	10,00
Shell thickness of stringers	10,00	10,00	10,00	10,00	10,00	10,00	10,00	15,00	20,00	10,00	10,00
Stiffener thickness in outer skin	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	15,00	20,00
Stiffener thickness in inner skin	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00
Stiffener thickness in web frame	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00
Stiffener thickness in stringers	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00
Built	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Run	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Postprocessing	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Comment											
Smallest element edge size	80.3	80,00	80,00	80,00	80,00	80,00	80,00	80,00	80,00	80,00	80,00
Fs	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29	
Weight	100,59	99,271	104,41	98,242	102,36	95,773	97,445	95,85	97,591	95,077	96,027
Increase	6,4635	5,1433	10,287	4,1147	8,2295	1,6459	3,3175	1,7227	3,4635	0,9495	1,8991
Energy absorbed at max indentation	50,52	59,55	86,11	47,23	55,02	41,59	39,63	42,46	45,56	43,27	43,49
Safe distance	2,86	2,31	2,1	3,07	2,69			3,63	3,36	3,54	3,52
Reduction in safe distance	0,82	1,37	1,58	0,61	0,99			0,05	0,32	0,14	0,16
Comparison ratio	0,1269	0,2664	0,1536	0,1482	0,1203			0,029	0,0924	0,1474	0,0843

Parameter matrix setup

Variables	19,00	20,00	21,00	22,00	23,00	24,00	25,00	26,00	27,00	28,00	29,00
Length between webframes	2100,00	2100,00	2100,00	2100,00	2100,00	2100,00	2100,00	2100,00	2100,00	2100,00	2100,00
Height between stringers	2600,00	2600,00	2600,00	2600,00	2600,00	2600,00	2600,00	2600,00	2600,00	2600,00	2600,00
Number of stiffeners in each stringer distance outer skin	3,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	5,00
Number of stiffeners in each stringer distance inner skin	3,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00
Shell thickness of outer skin	10,00	10,00	10,00	10,00	10,00	10,00	15,00	10,00	20,00	10,00	10,00
Shell thickness of inner skin	8,00	8,00	8,00	8,00	8,00	8,00	8,00	8,00	8,00	8,00	8,00
Shell thickness of web frame	10,00	10,00	10,00	10,00	10,00	10,00	15,00	15,00	20,00	20,00	10,00
Shell thickness of stringers	10,00	10,00	10,00	10,00	10,00	10,00	15,00	15,00	20,00	20,00	10,00
Stiffener thickness in outer skin	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00
Stiffener thickness in inner skin	15,00	20,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00
Stiffener thickness in web frame	10,00	10,00	15,00	20,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00
Stiffener thickness in stringers	10,00	10,00	10,00	10,00	15,00	20,00	10,00	10,00	10,00	10,00	10,00
Built	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Run	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Postprocessing	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Comment											
Smallest element edge size	80,00	80,00	80,00	80,00	80,00	80,00	80,00	80,00	80,00	80,00	80,00
Fs	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29
Weight	95,077	96,027	94,484	94,84	94,602	95,077	102,64	97,496	111,2	100,91	96,464
Increase	0,9495	1,8991	0,3561	0,7123	0,4748	0,9495	8,5115	3,3686	17,068	6,7815	2,3366
Energy absorbed at max indentation	43,12	43,92	38,21	37,84	45,86	46,25	65,49	41,38	79,78	42,94	48,22
Safe distance	3,55	3,46			3,23	3,17	2,16		1,88	3,59	2,95
Reduction in safe distance	0,13	0,22			0,45	0,51	1,52		1,80	0,09	0,73
Comparison ratio	0,1369	0,1158			0,9478	0,5371	0,1786		0,1055	0,0133	0,3124

Parameter matrix setup

Variables	30,00	31,00	32,00	Origina	ICELong	ICEVert
Length between webframes	2100,00	2100,00	2100,00	sfd	sfd	sfd
Height between stringers	2600,00	2600,00	2600,00	sfd	sfd	sfd
Number of stiffeners in each stringer distance outer skin	7,00	3,00	3,00	sfd	sfd	sfd
Number of stiffeners in each stringer distance inner skin	3,00	5,00	7,00	sfd	sfd	sfd
Shell thickness of outer skin	10,00	10,00	10,00	sfd	sfd	sfd
Shell thickness of inner skin	8,00	8,00	8,00	sfd	sfd	sfd
Shell thickness of web frame	10,00	10,00	10,00	sfd	sfd	sfd
Shell thickness of stringers	10,00	10,00	10,00	sfd	sfd	sfd
Stiffener thickness in outer skin	10,00	10,00	10,00	sfd	sfd	sfd
Stiffener thickness in inner skin	10,00	10,00	10,00	sfd	sfd	sfd
Stiffener thickness in web frame	10,00	10,00	10,00	sfd	sfd	sfd
Stiffener thickness in stringers	10,00	10,00	10,00	sfd	sfd	sfd
Built	Yes	Yes	Yes	No	No	No
Run	Yes	Yes	Yes	No	No	No
Postprocessing	Yes	Yes	Yes	No	No	No
Comment						
Smallest element edge size	*80,00*	*80,00*	*80,00*			
Fs	*0.29*	*0.29*	*0.29*			
Weight	98,059	96,468	98,0593	96,59	101,59	101,65
Increase	3,9318	2,3409	3,9318	0,00	5,01	5,07
Energy absorbed at max indentation	45,86	45,11	43,42	40,86	70,16	63,69
Safe distance	3,28	3,34	3,53	3,68	2,23	2,31
Reduction in safe distance	0,40	0,34	0,15		1,45	1,37
Comparison ratio	0,1017	0,1452	0,03815		0,2895	0,2704

E – MATLAB CODE

In this appendix the Matlab code is given, due to the length of the full code, only the functions which is concerned with the parametrical modelling is given in the written appendix. The full code is given in the electronic appendix.

input.txt

```
This file contains input data for a parametric ship section.

Written by Halvor Aga




Sections to be made:

(1 for first section, 0 for coarse, 2 for fine mesh, 3 for last and 4 for a course
section following a fine section)

1 0 2 2 2 2 4 3


General side section geometrical data:

g_wss  g_lbwf  g_Htt

 1200    2100    1250


Bow data:

R1    R2      L

1250 1937.5 4000


Deck structure data

Length between the longitudinal girders:

2400 2800 2800

Number of stiffeners between the longitudinal girders:

 3   3   3

Property numbers of the shell in deck surface between the longitudinal girders:

11 11 11

Property numbers of the plates in each of the longitudinal girders:

 9   9   9

Property numbers of the shell in transverse girder and the knuckle:

 8 10
```

Property numbers for stiffeners in deck surface

 8 8 8

Property numbers for stiffeners on longitudinal girders

 6 6 6

Property numbers for stiffeners on transverse girder and knuckle

 5 7

Side section data:

Height of each segment

2600 2600 2600

Number of stiffeners per stringer distance

 3 3 3

Property numbers for shell in outer skin

12 13 14

Property numbers for shell in inner skin

15 16 17

Property numbers for shell in web frame

18 20 22

Property numbers for shell with manhole in web frame

19 21 23

Property numbers for shell in tank top, stringers and deck

24 24 24 26

Property numbers for shell with manhole in tank top, stringers and deck

25 25 25 26

Property numbers for stiffeners in web frame

 9 9 9

Property numbers for stiffeners in tank top, stringers and deck

10 10 10

Property numbers for stiffeners in outer skin

11 12 13

Property numbers for stiffeners in inner skin

14 15 16

Width for manholes

 800

```
Double bottom data:

Width in each section

 2400  2800  2800

Number of stiffeners in section

 3  3  3

Property numbers for shell in outer skin

 7  7  7  7

Property numbers for shell at tanktop

 6  6  6

Property numbers for shell in web frame

 2  1  1  1

Properties for shell with manhole cutout and drillings in webframe

 3 2

Property numbers for stiffeners in web frame

 1  1  1

Property numbers for shell in longitudinal girders

 5  5  5  4

Property numbers for shell in longitudinal girders where manholes are present

 5  5  5  5

Property numbers for stiffeners in longitudinal girders

 2  2  2

Property numbers for stiffeners in outer skin

 4  4  4

Property numbers for stiffeners in tanktop

 3  3  3
```

sectioninput.txt

This file contains sections for use in modelgeneration

Shell sections

number of shell sections:

26

Section:	Name:	Material type:	Thickness:
db	wf	nlSteel	12.0
db	wfd	nlSteel	12.0
db	wfm	nlSteel	3.9
db	lg	nlSteel	12.0
db	lgm	nlSteel	3.9
db	tt	nlSteel	11.0
db	os	nlSteel	11.0
ds	tg	nlSteel	10.0
ds	lg	nlSteel	10.0
ds	kn	nlSteel	12.0
ds	dp	nlSteel	9.0
ss	os1	nlSteel	10.0
ss	os2	nlSteel	10.0
ss	os3	nlSteel	11.0
ss	is1	nlSteel	10.0
ss	is2	nlSteel	8.0
ss	is3	nlSteel	9.0
ss	wf1	nlSteel	12.0
ss	wfm1	nlSteel	3.6
ss	wf2	nlSteel	9.0
ss	wfm2	nlSteel	2.7
ss	wf3	nlSteel	10.0
ss	wfm3	nlSteel	3.0
ss	str	nlSteel	10.0
ss	strm	nlSteel	3.0
ss	dp	nlSteel	9.0

Beam sections

number of beam sections:

16

section: Name:Materal type: Thickness: Height: Direction vector (xyz): offset vector (xyz):

section:	Name:	Materal type:	Thickness:	Height:	Direction vector (xyz):			offset vector (xyz):		
db	wf	nlSteel	120	10	0	-1	0	60	0	0
db	lg	nlSteel	120	10	0	0	-1	0	-60	0
db	tt	nlSteel	200	11.8	0	-1	0	0	0	-100
db	os	nlSteel	180	10.5	0	1	0	0	0	90
ds	tg	nlSteel	200	20	0	0	-1	0	0	-10
ds	lg	nlSteel	200	20	0	0	-1	0	0	-10
ds	kn	nlSteel	100	12	0	1	-1	0	4.24	-4.24
ds	dp	nlSteel	120	8.75	0	-1	0	0	0	-60
ss	wf	nlSteel	120	10	0	0	-1	60	0	0
ss	str	nlSteel	120	10	0	-1	0	0	0	-60
ss	os1	nlSteel	180	10.5	0	0	-1	0	90	0
ss	os2	nlSteel	160	10.1	0	0	-1	0	80	0
ss	os3	nlSteel	140	10.9	0	0	-1	0	70	0
ss	is1	nlSteel	180	10.5	0	0	1	0	-90	0
ss	is2	nlSteel	160	10.1	0	0	1	0	-80	0
ss	is3	nlSteel	140	10.9	0	0	1	0	-70	0

Mesh sizes:

160 80

runscript.m

```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Script for making a series of convergence analysis executable in LS-DYNA%
%                                                                         %
% Written by Halvor L. Aga as part of the work concerning                 %
% "Assessement if structural requirements related to LNG fuel tanks."     %
% Deadline 10. June 2013                                                  %
% Has been made by modification of recorded .ses commands in PATRAN.      %
%                                                                         %
% Build-up:                                                               %
%       0. Check if parts of the study exists.                            %
%         1. Stop if it does.                                             %
%       1. Create study map structure.                                    %
%       2. Copy input files                                               %
%       3. Open files                                                     %
%       4. Create the .ses by function model_generation                   %
%       5. Manual intervation: Excecute the .ses files in PATRAN          %
%       6. Manipulate keyword files from PATRAN for boundary conditions,   %
%          beam offset and to implement necessary keyword commands as given%
%          in "keyword" file.                                             %
%       7. Write the location of the input file, as well as the location of%
%          the results file in a output file.                             %
%                                                                         %
% Input:                                                                  %
%      input.txt        - File containing input for the geometrical       %
%      sectioninput.txt - File containing input for sections and mesh     %
%                         sizes                                           %
%      keyword.txt      - File containing all necessary keyword commands   %
%                         for execution in LS-DYNA                        %
% Output:                                                                 %
%      Keyword files with analysis ready for execution in LS-DYNA         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Clearing variables
clear all;
clc;

rootfolder = cd;
studyname  = 'Original';
inputfilefolder = [cd '\Input_files'];

if exist([rootfolder '\' studyname],'file') == 0
        dbloc=[rootfolder '\' studyname '\database'];
        mkdir(dbloc);
        copyfile([inputfilefolder '\' 'input.txt'],dbloc)
        copyfile([inputfilefolder '\' 'sectioninput.txt'],dbloc)
        copyfile([inputfilefolder '\' 'keyword.txt'],dbloc)


        %Create file for .ses file storage
        fileID(1) =  fopen([dbloc '\' 'modelfile.ses'],'w');
        % Open input files
        fileID(2) =  fopen([dbloc '\' 'input.txt'],'r');
        fileID(3) =  fopen([dbloc '\' 'sectioninput.txt'],'r');
        fileID(4) =  fopen([dbloc '\' 'keyword.txt'],'r+');

        filepath = [dbloc '\' 'modelfile.ses'];
        % Generate session file
        [offset,bpname,WF,STR] = model_generation(fileID,filepath,dbloc);
        % Nesting of offset and byname
        offsetsaml(1,:) = offset(:,1);
        offsetsaml(2,:) = offset(:,2);
        offsetsaml(3,:) = offset(:,3);
        bpnamesaml(1,:) = bpname(:,1);
        for j = 1:4
            caseloc = [rootfolder '\' studyname '\' 'bow_pos_#' num2str(j)];
            mkdir(caseloc)
        end
```

```matlab
    loc = [rootfolder '\' studyname];
    postID = fopen([rootfolder '\' studyname '\' 'postscript.ses'],'w');
    postscript(WF,STR,loc,postID,studyname);

    [dbloc '\' 'modelfile.ses']

    % Run session files manually in PATRAN
    reply = input('Press Y when session file listed over is been manually executed
by PATRAN (0 to abort):\n','s');

    if reply == 'Y'
        % Create file for storage of work file locations
        workID = fopen([rootfolder '\' studyname '\' 'workfile'],'w');
            dbloc=[rootfolder '\' studyname '\database'];
            for j = 1:4
                caseloc = [rootfolder '\' studyname '\' 'bow_pos_#' num2str(j)];
                filename = [dbloc '\' 'bulb' num2str(j) '.key'];
                addfilename = [dbloc '\' 'keyword.txt'];
                finishfilename = [caseloc '\' 'finishedfile.key'];
                % Un-nesting of offset and bpname
                offset(:,1) = offsetsaml(1,:);
                offset(:,2) = offsetsaml(2,:);
                offset(:,3) = offsetsaml(3,:);
                bpname(:,1) = bpnamesaml(1,:);
                % Add keywords to keyword file from PATRAN
                keyman(filename,addfilename,finishfilename,offset,bpname);
                % Store location of the finished keyword file to work list
                workfile = [finishfilename];
                % Store location of work list
                fprintf(workID, '%s\n', workfile );
            end
        fprintf('workfilenames and output folders written to files in studymap\n')
    else
        fprintf('Excecution aborted\n')
    end
else
   fprintf('Folder exists, nothing has been done. Specify unique name and rerun\n')
end

fclose('all');


% End of script
```

model_generation.m

```matlab
function [offsetm,bpname,WF,STR] = model_generation(filesID,filepath1,...
    filepath2)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function generates the models needed to create a full ship section %
% for analysis in LS-DYNA by the use of a PATRAN session file.            %
%                                                                         %
% Written by Halvor L. Aga as part of the work concerning                 %
% "Assessement if structural requirements related to LNG fuel tanks."     %
% Deadline 10. June 2013                                                   %
% Has been made by modification of recorded .ses commands in PATRAN.      %
%                                                                         %
% Build-up:                                                               %
%       1. Read input data from file                                      %
%       2. Write necessary commands to a .ses file for creation of:       %
%         1. Ship section geometry                                        %
%         3. Meshing                                                      %
%         2. Bulbs                                                        %
%         4. Analysis commands (commands to make PATRAN make  .key        %
%             files for each of the bow positions compatible with LS-DYNA) %
%                                                                         %
% Input:                                                                  %
%       filesID: Vector with file identifiers for files used             %
%       filepath1: File path to location of files written including .ses  %
%                 filename                                                %
%       filepath2: File path to location of files written                %
%       filename: The name of the executable .ses file                    %
%                                                                         %
% Output:                                                                 %
%       offsetm: Matrix containing data for creating beam offset          %
%       bpname: Vector including ordered names for creating beam offset    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%
% Read input data from file:

% Open input file:
inputID = filesID(2);
for i = 1:4 % Omitting lines
    fgets(inputID);
end

% Read filename (not used)
fgetl(inputID);

% Read input data:
for i =1:3 % Omitting lines
    fgets(inputID);
end

% Sections to be made
section = strread(fgets(inputID));

for i =1:3 % Omit lines
    fgets(inputID);
end
WF = length(section)-1;

% General data:
[g_wss g_lbwf g_Htt] = strread(fgets(inputID));

for i =1:3 % Omit lines
    fgets(inputID);
end

% Bow data
[R1 R2 L] = strread(fgets(inputID));
```

```matlab
for i =1:3 % Omit lines
    fgets(inputID);
end

% Deck structure data:
% Width of each section                    [mm]
ds_W  = strread(fgets(inputID));
fgets(inputID);
% Number of stiffeners in each section     [-]
ds_ns = strread(fgets(inputID));
fgets(inputID);
% Property numbers for shell in deck surface
ds_plt_ds = strread(fgets(inputID));
fgets(inputID);
% Property numbers for shell in longitudinal girders
ds_plt_lg = strread(fgets(inputID));
fgets(inputID);
% Property numbers for shell in transverse girder and knuckle
ds_plt_tg = strread(fgets(inputID));
fgets(inputID);
% Property numbers for stiffeners in deck surface
ds_sti_ds = strread(fgets(inputID));
fgets(inputID);
% Property numbers for stiffeners on longitudinal girders
ds_sti_lg = strread(fgets(inputID));
fgets(inputID);
% Property numbers for stiffeners on transverse girder and knuckle
ds_sti_tg = strread(fgets(inputID));

for i =1:3 % Omitting lines
    fgets(inputID);
end

% Side section data:
% Height of each segment                    [mm]
ss_H     = strread(fgets(inputID));
fgets(inputID);
% Number of stiffeners per stringer distance[-]
ss_nst   = strread(fgets(inputID));
fgets(inputID);

STR = length(ss_nst)-1;
% Property numbers for shell in outer skin
ss_plt_os  = strread(fgets(inputID));
fgets(inputID);
% Property numbers for shell in inner skin
ss_plt_is  = strread(fgets(inputID));
fgets(inputID);
% Property numbers for shell in web frame
ss_plt_wf(1,:)  = strread(fgets(inputID));
fgets(inputID);
% Property numbers for shell with manholes in web frame
ss_plt_wf(2,:)  = strread(fgets(inputID));
fgets(inputID);
% Property numbers for shell in tank top, stringers and deck
ss_plt_str(1,:) = strread(fgets(inputID));
fgets(inputID);
% Property numbers for shell with manholes in tank top, stringers and deck
ss_plt_str(2,:) = strread(fgets(inputID));
fgets(inputID);
% Property numbers for stiffeners in web frame
ss_sti_wf  = strread(fgets(inputID));
fgets(inputID);
% Property numbers for stiffeners in tank top, stringers and deck
ss_sti_str = strread(fgets(inputID));
fgets(inputID);
% Property numbers for stiffeners in outer skin
```

```matlab
ss_sti_os  = strread(fgets(inputID));
fgets(inputID);
% Property numbers for stiffeners in inner skin
ss_sti_is  = strread(fgets(inputID));

for i =1:2 % Omit lines
    fgets(inputID);
end

% Data for manholes
wmh =strread(fgets(inputID)); % Width

for i =1:3 % Omit lines
    fgets(inputID);
end

% Double bottom data:
% Width in each section                          [mm]
db_W = strread(fgets(inputID));
fgets(inputID);
% Number of stiffeners in section 1, 2 and 3 [-]
db_ns = strread(fgets(inputID));
fgets(inputID);
% Property numbers for shell in outer skin
db_plt_os  =strread(fgets(inputID));
fgets(inputID);
% Property numbers for shell at tanktop
db_plt_tt  =strread(fgets(inputID));
fgets(inputID);
% Property numbers for shell in web frame
db_plt_wf(1,:)  = strread(fgets(inputID));
fgets(inputID);
% Properties for shell with manhole cutout and drillings
db_plt_wf_s =strread(fgets(inputID));
fgets(inputID);
% Property numbers for stiffeners in web frame
db_sti_wf  = strread(fgets(inputID));
fgets(inputID);
% Property numbers for shell in longitudinal girders
db_plt_str(1,:) = strread(fgets(inputID));
fgets(inputID);
% Property numbers for reduced shell in longitudinal girders
db_plt_str(2,:) = strread(fgets(inputID));
fgets(inputID);
% Property numbers for stiffeners in longitudinal girders
db_sti_str = strread(fgets(inputID));
fgets(inputID);
% Property numbers for stiffeners in outer skin
db_sti_os  = strread(fgets(inputID));
fgets(inputID);
% Property numbers for stiffeners in tanktop
db_sti_tt  = strread(fgets(inputID));
%%

hg = 550;   % Height of transverse girder
hknl = 600; % Height dimension of knuckle
% Create counters used in the whole program
sc = 0; % Surface counter

% Create matrices for storage of meshing data
fem_dat_sh = 0; % Shell element data
fem_dat_be = 0; % Beam element data
bc=0; % Boundary condition data

% Session file generation:
% Create file and setting viewport
initialize(filesID(1),filepath1,filepath2);
% Creating ship section
```

```matlab
for i = 1:length(section)

    % Offset in x-direction
    xof = g_lbwf*(i-length(section)/2-1);

    % Write commands for deck structure to .ses file
    [sc,fem_dat_sh,fem_dat_be,bc] = deck(filesID(1),xof,g_wss,g_lbwf,....
    ss_H,ds_ns,ds_W,sc,section(i),ds_plt_ds,ds_plt_lg,ds_plt_tg,ds_sti_ds,...
    ds_sti_lg,ds_sti_tg,fem_dat_sh,fem_dat_be,bc,hg,hknl);

    % Write commands for side section to .ses file
    [sc,fem_dat_sh,fem_dat_be,bc] = side_section(filesID(1),xof,g_wss,...
        g_lbwf,ss_H,ss_nst,sc,ss_plt_os,ss_plt_is,ss_plt_wf,ss_sti_wf,...
        ss_plt_str,ss_sti_str,ss_sti_os,ss_sti_is,fem_dat_sh,fem_dat_be,...
        section(i),bc);

    % Write commands for double bottom to .ses file
    [sc,fem_dat_sh,fem_dat_be,bc] = double_bottom(filesID(1),xof,g_lbwf,...
        g_Htt,db_ns,db_W,g_wss,sc,wmh,section(i),db_plt_os,db_plt_tt,...
        db_plt_wf,db_sti_wf,db_plt_str,db_sti_str,db_sti_os,db_sti_tt,...
        fem_dat_sh,fem_dat_be,db_plt_wf_s,bc);
end

% Create meshing
[offsetm,bpname,finemesh]=mesh(filesID(1),fem_dat_sh,fem_dat_be,bc,filesID);
% Calculate bow positions
bpos = [0 -101  ss_H(1)/2;
    0 -101  ss_H(1);
    g_lbwf/2 -101  ss_H(1);
    g_lbwf/2 -101  ss_H(1)+ss_H(2)/2];
% Save without bulb
nobulb = [filepath2 '\nobulb.db'];
fprintf(filesID(1),'uil_file_saveas.copy( "%s", FALSE )\n',nobulb);
fprintf(filesID(1),'uil_file_close.go(  )\n');
% Create keyword file for each bow position
for i = 1:4
    % Open shipside model
    fprintf(filesID(1),'uil_file_open.go( "%s" )\n',nobulb);
    % Save with new filename
    abulb = [filepath2 '\bulb' num2str(i) '.db'];
    fprintf(filesID(1),'uil_file_saveas.copy( "%s", FALSE )\n',abulb);
    % Open new file
    fprintf(filesID(1),'uil_file_close.go(  )\n');
    fprintf(filesID(1),'uil_file_open.go( "%s" )\n',abulb);
    % Create bulb
    bulb(filesID(1),R1,R2,L,bpos(i,3),bpos(i,2),bpos(i,1),sc,2*finemesh);
    % Create keyword file
    analysis(filesID(1),['bulb' num2str(i)],filepath2);
end

end
```

bulb.m

```matlab
function [sc] = bulb(fileID,R1,R2,L,zoff,yoff,xoff,sc,esize)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function generates necessary commands for generation of a bow in PATRAN   %
%                                                                               %
% Written by Halvor L. Aga as part of the work concerning                       %
% "Assessement if structural requirements related to LNG fuel tanks."           %
% Deadline 10. June 2013                                                        %
% Has been made by modification of recorded .ses commands in PATRAN.            %
%                                                                               %
% Input:                                                                        %
%        fileID - identification key for keyword file                           %
%        Scantlings (R1 - Radius at striking end of the bow, R2 - Radius at     %
%                    following end of bow, L - Length of bow)                    %
%        Offset from origo (zoff,toff,xoff)                                      %
%        sc     - surface counter                                               %
%        esize  - mesh size on the bulb                                         %
%                                                                               %
% Output: sc     - surface counter                                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% Checking if length of bow and radius given as input are compatible.
if L>R1
    % Printing commands for point generation
    fprintf(fileID,'STRING asm_create_grid_xyz_created_ids[VIRTUAL]\n');
    fprintf(fileID,'asm_const_grid_xyz( "1", "[%d %d %d]", @\n',xoff,yoff,zoff);
    fprintf(fileID,'"Coord 0", asm_create_grid_xyz_created_ids )\n');
    fprintf(fileID,'asm_const_grid_xyz( "2", "[%d %f %f]", @\n',xoff,-R1-yoff,-
R1+zoff);
    fprintf(fileID,'"Coord 0",  asm_create_grid_xyz_created_ids )\n');
    fprintf(fileID,'asm_const_grid_xyz( "3", "[%d %f %f]",  @\n',xoff,-L-yoff,-
R2+zoff);
    fprintf(fileID,'"Coord 0", asm_create_grid_xyz_created_ids )\n');
    fprintf(fileID,'asm_const_grid_xyz( "4", "[%d %f %d]",  @\n',xoff,-L-
yoff,zoff);
    fprintf(fileID,'"Coord 0", asm_create_grid_xyz_created_ids )\n');

    % Printing commands for curve generation
    fprintf(fileID,'STRING sgm_create_curve_2d_created_ids[VIRTUAL]\n');
    fprintf(fileID,'sgm_const_curve_2d_arc2point_v2( "1", 2, %f, FALSE, TRUE, 1,
@\n',R1);
    fprintf(fileID,'"Coord 0.1", "", "point 1", "point 2", FALSE,@\n');
    fprintf(fileID,'sgm_create_curve_2d_created_ids )\n');
    fprintf(fileID,'STRING asm_line_2point_created_ids[VIRTUAL]\n');
    fprintf(fileID,'asm_const_line_2point( "2", "point 2", "point 3", 0, "", 50.,
1,  @\n');
    fprintf(fileID,'asm_line_2point_created_ids )\n');
    fprintf(fileID,'asm_const_line_2point( "4", "point 4", "point 1", 0, "", 50.,
1,  @\n');
    fprintf(fileID,'asm_line_2point_created_ids )\n');


    sc = sc+1;

    % Printing commands for solid generation
    fprintf(fileID,'STRING sgm_sweep_surface_r_created_ids[VIRTUAL]\n');
    fprintf(fileID,'sgm_const_surface_revolve( "%d", "Construct
CurvePointTangent(Evaluate Geometry(Curve 4))(Evaluate Geometry(Point 1))", 360.,
0., "Coord 0", "Curve 1:2", sgm_sweep_surface_r_created_ids )\n',sc);
    % Printing commands for material generation
    fprintf(fileID,'material.create( "Analysis code ID", 10001, "Analysis type ID",
1,  @\n');
    fprintf(fileID,'"Rigid_steel", 0, " ", "Isotropic", 1,  @\n');
    fprintf(fileID,'"Directionality", 1, "Linearity", 11001, "Homogeneous", 0,
"Rigid", 11001,  @\n');
```

```matlab
    fprintf(fileID,'"Model Options & IDs", ["Material Type 20", "", "", "", ""],
[11006, 0, 0, 0,  @\n');
    fprintf(fileID,'0], "Active Flag", 1, "Create", 10, "External Flag", FALSE,
"Property IDs", [ @\n');
    fprintf(fileID,'"Density", "Elastic Modulus", "Poisson Ratio"], [16, 2, 5, 0],
@\n');
    fprintf(fileID,'"Property Values", ["7.85e-9", "2.07e5", "0.3", ""] )\n');
    fprintf(fileID,'elementprops_create( "Indenter", 71, 25, 20, 11027, 1, 20, [13,
20, 1011,  @\n');
    fprintf(fileID,'11182], [5, 2, 1, 1], ["m:Rigid_steel", "", "", ""], )\n');
    % Printing commands for property generation
    fprintf(fileID,'elementprops_create( "%s",@\n',char('Indenter'));
    fprintf(fileID,'51, 25, 35, 11004, 1, 20,@\n');
    fprintf(fileID,'[13, 20, 36, 1004, 11044, 11136, 11027, 1011, 11182],@\n');
    fprintf(fileID,'[5, 2, 1, 1, 4, 3, 4, 1, 1],@\n');
    fprintf(fileID,'["m:%s", "", "%d", "0.833", "", "5", "", "",
""],@\n',char('Rigid_steel'),2);
    fprintf(fileID,'"" )\n');

    % Printing commands for mesh generation
    fprintf(fileID,'INTEGER fem_create_mesh_surfa_num_nodes\n');
    fprintf(fileID,'INTEGER fem_create_mesh_surfa_num_elems\n');
    fprintf(fileID,'STRING fem_create_mesh_s_nodes_created[VIRTUAL]\n');
    fprintf(fileID,'STRING fem_create_mesh_s_elems_created[VIRTUAL]\n');
    fprintf(fileID,'fem_create_mesh_surf_4( "Hybrid", 49664,@\n');
    fprintf(fileID,'"Surface %d:%d", 4, ["%d", "0.1", "0.2",
"1.0"],@\n',sc,sc+1,esize);
    fprintf(fileID,'"Quad4", "#", "#", "Coord 0", "Coord 0",@\n');
    fprintf(fileID,'fem_create_mesh_surfa_num_nodes,@\n');
    fprintf(fileID,'fem_create_mesh_surfa_num_elems,@\n');
    fprintf(fileID,'fem_create_mesh_s_nodes_created,@\n');
    fprintf(fileID,'fem_create_mesh_s_elems_created )\n');
    fprintf(fileID,'fem_associate_elems_to_ep( "%s",@\n',char('Indenter'));
    fprintf(fileID,'fem_create_mesh_s_elems_created,@\n');
    fprintf(fileID,'fem_create_mesh_surfa_num_elems )\n');

else
    % Printing error message to screen
    fprintf('faulty input, radius bigger than length, no bulb has been made\n')
end

end
```

deck.m

```matlab
function [sc,fem_dat_sh,fem_dat_be,bc] = deck(fileID,xof,g_wss,g_lbwf,....
    ss_H,ds_ns,ds_W,sc,section,ds_plt_ds,ds_plt_lg,ds_plt_tg,ds_sti_ds,...
    ds_sti_lg,ds_sti_tg,fem_dat_sh,fem_dat_be,bc,hg,hknl)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Function for generation of PATRAN session file commands which will     %
% generate a deck section when run.                                     %
%                                                                       %
% Written by Halvor L. Aga as part of the work concerning                %
% "Assessement if structural requirements related to LNG fuel tanks."   %
% Deadline 10. June 2013                                                %
% Has been made by modification of recorded .ses commands in PATRAN.    %
%                                                                       %
% Input:                                                                %
% fileID - file identification number of the file for writing           %
% xof - Offset in x direction for the section                           %
% g_wss - Width of the side section                                     %
% g_lbwf - Length between the web frames                                %
% ss_H - Vertical distance between the stringers                        %
% ds_ns - number of stiffeners in each distance between the longitudinal %
%         girders                                                       %
% ds_W - Width between each of the longitudinal girders                 %
% sc - Counter for number of created surfaces                           %
% section - Information of placement of the section                     %
% ds_plt_ds - property numbers for plating in deck surface              %
% ds_plt_lg - property numbers for plating in longitudinal girders      %
% ds_plt_tg - property numbers for plating in transversal girders       %
% fem_dat_sh - matrix containing data regarding shell elements          %
% fem_dat_be - matrix containing data regarding beam elements           %
% bc - matrix containing data regarding boundary conditions             %
%                                                                       %
% Output:                                                               %
% sc - Counter for number of created surfaces                           %
% fem_dat_sh - matrix containing data regarding shell elements          %
% fem_dat_be - matrix containing data regarding beam elements           %
% bc - matrix containing data regarding boundary conditions             %
%                                                                       %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Determination of femc_sh, finite elements counter for shell elements
if (size(fem_dat_sh,2)==1); femc_sh = 0;
else femc_sh = size(fem_dat_sh,1);
end
% Determination of femc_be, finite elements counter for beam elements
if (size(fem_dat_be,2)==1); femc_be = 0;
else femc_be = size(fem_dat_be,1);
end
% Determination of bcc, counter for boundary conditions
if (size(bc,2)==1); bcc = 0;
else bcc = size(bc,1);
end

xofm = [xof, xof+(g_lbwf-800)/2, xof+(g_lbwf+800)/2];
lbwfm = [(g_lbwf-800)/2, 800, (g_lbwf-800)/2];

for b = 1:3
    i_xof = xofm(b);
    i_l = lbwfm(b);

%-----------------------------------------------------------------------
% Point generation

fprintf(fileID,'STRING asm_create_grid_xyz_created_ids[VIRTUAL]\n');

% Points on deck surface
y = g_wss;              % Horizontal position of the point.
g_Hmd = sum(ss_H);      % Height of the main deck in absolute coordinates
```

```matlab
pc = 1;                    % point counter
fprintf(fileID,'asm_const_grid_xyz( "%d", "[%d %d %d]", "Coord 0",  @\n'...
    ,pc,i_xof,y,g_Hmd);
fprintf(fileID,'asm_create_grid_xyz_created_ids )\n');

sco = sc; % Storing number of existing surfaces

for j = 1:length(ds_ns)
    for i = 1:ds_ns(j)+1
        pc = pc + 1;
        y = y + ds_W(j)/(ds_ns(j)+1);
        fprintf(fileID,...
            'asm_const_grid_xyz( "%d", "[%d %d %d]", "Coord 0",  @\n'...
            ,pc,i_xof,y,g_Hmd);
        fprintf(fileID,'asm_create_grid_xyz_created_ids )\n');
    end
end

% Points for knuckle
pc =pc+1;
fprintf(fileID,'asm_const_grid_xyz( "%d", "[%d %d %d]", "Coord 0",  @\n'...
    ,pc,i_xof,g_wss,g_Hmd-hg-hknl);
fprintf(fileID,'asm_create_grid_xyz_created_ids )\n');
pc =pc+1;
fprintf(fileID,'asm_const_grid_xyz( "%d", "[%d %d %d]", "Coord 0",  @\n'...
    ,pc,i_xof,g_wss,g_Hmd-hg);
fprintf(fileID,'asm_create_grid_xyz_created_ids )\n');
pc =pc+1;
fprintf(fileID,'asm_const_grid_xyz( "%d", "[%d %d %d]", "Coord 0",  @\n'...
    ,pc,i_xof,g_wss+hknl,g_Hmd-hg);
fprintf(fileID,'asm_create_grid_xyz_created_ids )\n');

% Point generation finished
%-------------------------------------------------------------------------
% Curve generation

% Curves on deck surface
cc = 0; % Curve counter
fprintf(fileID,'STRING asm_line_2point_created_ids[VIRTUAL]\n');
for i = 1:sum(ds_ns)+length(ds_ns)
    cc = cc + 1;
    fprintf(fileID,'asm_const_line_2point( "%d", "point %d", @\n',i,i);
    fprintf(fileID,'"point %d", 0, "", 50., 1,  @\n',i+1);
    fprintf(fileID,'asm_line_2point_created_ids )\n');
end

% Curve generation finished
%-------------------------------------------------------------------------
% Surface generation

% Web of transverse girder

fprintf(fileID,'STRING sgm_sweep_surface_e_created_ids[VIRTUAL]\n');
for i = 1:sum(ds_ns)+length(ds_ns)
    sc = sc+1;
    fprintf(fileID,'sgm_const_surface_extrude( @\n');
    fprintf(fileID,...
        '"%d", "<0 0 %d>", 1., 0., "[0 0 0]", "Coord 0", @\n',sc,-hg);
    fprintf(fileID,' "Curve %d", sgm_sweep_surface_e_created_ids )\n',i);
    if section ~= 1 && b == 1
        % Store finite elements data for surface
        femc_sh =femc_sh + 1;
        fem_dat_sh(femc_sh,1) = sc;              % Surface number
        fem_dat_sh(femc_sh,2) = ds_plt_tg(1);    % Property number
        fem_dat_sh(femc_sh,3) = 1;          % Mesh fineness
        % Store finite elements data for curve
        femc_be = femc_be +1;
        fem_dat_be(femc_be,1) = sc;              % Surface number
```

```matlab
            fem_dat_be(femc_be,2) = 2;                % Curve side number
            fem_dat_be(femc_be,3) = ds_sti_tg(1);    % Property number
            fem_dat_be(femc_be,4) = 1;               % Mesh fineness
        end
    end


    % Knuckle surface
    sc = sc+1;
    fprintf(fileID,'STRING sgm_create_surface__created_ids[VIRTUAL]\n');
    fprintf(fileID,'sgm_const_surface_vertex( @\n');
    fprintf(fileID,'"%d", "Point %d", "Point %d", "Point %d", "Point %d", @\n'...
        ,sc,pc-2,pc-1,pc,pc);
    fprintf(fileID,'sgm_create_surface__created_ids )\n');
    if section ~= 1 && b == 1
        femc_sh = femc_sh + 1;
        fem_dat_sh(femc_sh,:) = [sc ds_plt_tg(2) 1];
        femc_be = femc_be +1;
        fem_dat_be(femc_be,:) = [sc 4 ds_sti_tg(2) 1];
    end

    % Extrude longitudinal surfaces to one section
    fprintf(fileID,'STRING sgm_sweep_surface_e_created_ids[VIRTUAL]\n');
    count = 0;
    for i = 1:length(ds_ns)
        for j = 1:ds_ns(i)+1
            count = count + 1;
            sc = sc + 1;
            fprintf(fileID,'sgm_const_surface_extrude( @\n');
            fprintf(fileID,...
                '"%d", "<%d 0 0>", 1., 0., "[0 0 0]", "Coord 0", @\n',sc,i_l);
            fprintf(fileID,...
                ' "Curve %d", sgm_sweep_surface_e_created_ids )\n',count);
            femc_sh =femc_sh + 1;
            fem_dat_sh(femc_sh,1) = sc;
            fem_dat_sh(femc_sh,2) = ds_plt_ds(i);
            fem_dat_sh(femc_sh,3) = 1;
            % Create boundary conditions only if section is an end section or
            % the curve lies along the midline of the ship
            if section == 1 && b == 1
                bcc = bcc + 1;
                bc(bcc,1) = sc; % Surface number
                bc(bcc,2) = 4;  % Curve side number
                bc(bcc,3) = 1;  % Boundary conditions type
            elseif section == 3 && b == 3
                bcc = bcc + 1;
                bc(bcc,1) = sc;
                bc(bcc,2) = 2;
                bc(bcc,3) = 1;
            end
            % Create webframes if the section is not the first section
            if j~=1
                femc_be = femc_be +1;
                fem_dat_be(femc_be,1) = sc;
                fem_dat_be(femc_be,2) = 1;
                fem_dat_be(femc_be,3) = ds_sti_ds(i);
                fem_dat_be(femc_be,4) = 1;
            end
        end
    end

    % Longitudinal girders
    for i = 1 : length(ds_ns)
        sc = sc + 1;
        fprintf(fileID,'sgm_const_surface_extrude( @\n');
        fprintf(fileID,'"%d", "<%d 0 0>", 1., 0., "[0 0 0]", "Coord 0", @\n'...
            ,sc,i_l);
        fprintf(fileID,...
```

```matlab
        ' "Surface %d.3", sgm_sweep_surface_e_created_ids )\n'...
        ,sum(ds_ns(1:i))+i+sco);
    femc_sh =femc_sh + 1;
    fem_dat_sh(femc_sh,:) = [sc ds_plt_lg(i) 1];
    femc_be = femc_be +1;
    fem_dat_be(femc_be,:) = [sc,3,ds_sti_lg(i) 1];
     if section == 1 && b == 1
            bcc = bcc + 1;
            bc(bcc,1) = sc;  % Surface number
            bc(bcc,2) = 4;   % Curve side number
            bc(bcc,3) = 1;   % Boundary conditions type
        elseif section == 3 && b == 3
            bcc = bcc + 1;
            bc(bcc,1) = sc;
            bc(bcc,2) = 2;
            bc(bcc,3) = 1;
        end
end

% Surface generation finished
%-------------------------------------------------------------------------
% Cleaning

% Delete points and curves used
fprintf(fileID,'STRING asm_delete_point_deleted_ids[VIRTUAL]\n');
fprintf(fileID,...
    'asm_delete_point( "Point 1:3000", asm_delete_point_deleted_ids )\n');
fprintf(fileID,'STRING asm_delete_curve_deleted_ids[VIRTUAL]\n');
fprintf(fileID,...
    'asm_delete_curve( "Curve 1:100", asm_delete_curve_deleted_ids )\n');

% Cleaning finished
%-------------------------------------------------------------------------
end
end
```

side_section.m

```matlab
function [sc,fem_dat_sh,fem_dat_be,bc] = side_section(fileID,xof,g_wss,...
    g_lbwf,ss_H,ss_nst,sc,ss_plt_os,ss_plt_is,ss_plt_wf,ss_sti_wf,...
    ss_plt_str,ss_sti_str,ss_sti_os,ss_sti_is,fem_dat_sh,fem_dat_be,...
    section,bc)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function generates the neccessary session file commands to create  %
% a sidesection of a hull between two web frames by the use of a PATRAN    %
% session file.                                                           %
%                                                                        %
% Written by Halvor L. Aga as part of the work concerning                 %
% "Assessement if structural requirements related to LNG tanks."          %
% Deadline on 10. June 2013                                               %
% Has been made by modification of recorded .ses commands in PATRAN.      %
%                                                                        %
% Build-up:                                                               %
% 1. Determintation of counters used for storage of finite elements and   %
%    boundary conditions data.                                           %
% 2. Determintation of offsets and lengths of section before manhole, with%
%    manhole and after manhole                                           %
% 3. Generation of geometry and storage of FEM data and BC data. (Done in %
%    loop to create separate platefields where manholes exist.)           %
%    1. Generation of curves for extrusion                               %
%    2. Extrusion of:                                                    %
%       1. Web frame                                                     %
%       2. Inner skin                                                    %
%       3. Outer skin                                                    %
%       4. Stringers                                                     %
%       5. Deck                                                          %
%    3. Deletion of temporary curves and points                          %
%                                                                        %
% For input and output descriptions see model_generation.m                %
%                                                                        %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Determinate femc_sh, finite elements counter for shell elements
if (size(fem_dat_sh,2)==1); femc_sh = 0;
else femc_sh = size(fem_dat_sh,1);
end
% Determinate femc_be, finite elements counter for beam elements
if (size(fem_dat_be,2)==1); femc_be = 0;
else femc_be = size(fem_dat_be,1);
end
% Determinate bcc, counter for boundary conditions
if (size(bc,2)==1); bcc = 0;
else bcc = size(bc,1);
end
% Calculate offset and length of each subsection
xofm = [xof, xof+(g_lbwf-800)/2, xof+(g_lbwf+800)/2];
lbwfm = [(g_lbwf-800)/2, 800, (g_lbwf-800)/2];

% Generate three sub sections
for b = 1:3
    % Chose lenght and offset of current subsection
    i_xof = xofm(b);
    i_l = lbwfm(b);
    lb = 0; % Switch for breaking of top plating of inner skin
    % Input generation for points
    fprintf(fileID,'STRING asm_create_grid_xyz_created_ids[VIRTUAL]\n');
    % Create bottom points
    z = 0;
    pc = 1; % Point counter
    fprintf(fileID,'asm_const_grid_xyz( "%d", "[%d %d %d]",  @\n',...
        pc,i_xof,0,z);
    fprintf(fileID,'"Coord 0", asm_create_grid_xyz_created_ids )\n');
    for i = 1:length(ss_H) % Number of plate fields in height direction
        for j = 1:ss_nst(i)+1 % Number of stiffener spacings in each
```

```matlab
            % Calculate point numbers and vertical position
            pc = pc + 1;
            z = z + ss_H(i)/(ss_nst(i)+1);
            % Write commands to file
            fprintf(fileID,...
                'asm_const_grid_xyz( "%d", "[%d %d %d]",  @\n',...
                pc,i_xof,0,z);
            fprintf(fileID,...
                '"Coord 0", asm_create_grid_xyz_created_ids )\n');
    end
end

% Create curves for extrusion
fprintf(fileID,'STRING asm_line_2point_created_ids[VIRTUAL]\n');
cc = 0; % Create curve counter
for i = 1:pc-1
    cc = cc+1;
    fprintf(fileID,'asm_const_line_2point( "%d", "point %d",@\n',i,i);
    fprintf(fileID,'"point %d", 0, "", 50., 1,  @\n',i+1);
    fprintf(fileID,'asm_line_2point_created_ids )\n');
end

count = 0; % Counter for determination of curve number
zof = 0;    % Variable for z offset of eac section in vertical direction
topmesh = 0; % Variable for creating fine mesh on top of this area
% Extrude to form
for i = 1:length(ss_nst)
    if i>=2; zof = sum(ss_H(1:i-1)); end
    for j = 1 : ss_nst(i)+1
        % web frame
        % Choose mesh fineness
        if section == 2 && zof <= 5000 && topmesh == 0
            mesh = 2;
        elseif topmesh == 0 && section == 2
            topmesh = 2;
            mesh = 1;
        else
            topmesh = 1;
            mesh = 1;
        end
        % Variable to get finemesh on end of section
        if section == 4 && zof <= 5000
            endmesh = 2;
        else
            endmesh = mesh;
        end
        sc = sc + 1;
        count = count + 1;
        % Extrude commands
        fprintf(fileID,...
            'STRING sgm_sweep_surface_e_created_ids[VIRTUAL]\n');
        fprintf(fileID,'sgm_const_surface_extrude( "%d",@\n',sc);
        fprintf(fileID,...
            '"<0 %d 0>", 1., 0., "[0 0 0]", "Coord 0",  @\n',g_wss);
        fprintf(fileID,...
            '"Curve %d", sgm_sweep_surface_e_created_ids )\n',count);
        % Breake web frame for easier meshing
        coord = [i_xof, (g_wss-650)/2, zof+(j-1)/(ss_nst(i)+1)*ss_H(i);
            i_xof,g_wss-(g_wss-650)/2 ,zof+(j-1)/(ss_nst(i)+1)*ss_H(i);
            i_xof, (g_wss-650)/2,        zof+j/(ss_nst(i)+1)*ss_H(i);
            i_xof, g_wss-(g_wss-650)/2,zof+j/(ss_nst(i)+1)*ss_H(i)];
        sc = prep_string(fileID,coord,sc,1,sc);
        % Save fem data, webframes ignored if it is the first section
        if section ~=1 && b == 1;
            for k = 1:3
                femc_sh = femc_sh + 1;
                fem_dat_sh(femc_sh,1) = sc-k+1;
                if (j-1)/(ss_nst(i)+1)*ss_H(i) <= 800
```

```matlab
                    fem_dat_sh(femc_sh,2) = ss_plt_wf(2,i);
                else
                    fem_dat_sh(femc_sh,2) = ss_plt_wf(1,i);
                end
                fem_dat_sh(femc_sh,3) = endmesh;
                if j~=1
                    femc_be = femc_be +1;
                    % Surface number
                    fem_dat_be(femc_be,1) = sc-k+1;
                    % Curve side number
                    fem_dat_be(femc_be,2) = 1;
                    % Property number
                    fem_dat_be(femc_be,3) = ss_sti_wf(1,i);
                    % Mesh fineness
                    fem_dat_be(femc_be,4) = endmesh;
                end
            end
        end

        sc = sc + 1;
        % inner skin
        fprintf(fileID,'sgm_const_surface_extrude( "%d",@\n',sc);
        fprintf(fileID,...
            '"<%d 0 0>", 1., 0., "[0 0 0]", "Coord 0",  @\n',i_l);
        fprintf(fileID,...
            '"Surface %d.2", sgm_sweep_surface_e_created_ids )\n',sc-3);
        femc_sh = femc_sh + 1;
        fem_dat_sh(femc_sh,1) =  sc;
        fem_dat_sh(femc_sh,2) = ss_plt_is(i);
        fem_dat_sh(femc_sh,3) = mesh;
        % Save boundary condition data if these apply
        if section == 1 && b == 1
            bcc = bcc + 1;
            bc(bcc,1) = sc;
            bc(bcc,2) = 4;
            bc(bcc,3) = 1;
        elseif section == 3 && b == 3
            bcc = bcc + 1;
            bc(bcc,1) = sc;
            bc(bcc,2) = 2;
            bc(bcc,3) = 1;
        end
        % Save FEM data
        if j~=1
            femc_be = femc_be +1;
            fem_dat_be(femc_be,1) = sc; % Surface number
            fem_dat_be(femc_be,2) = 1;   % Curve side number
            fem_dat_be(femc_be,3) = ss_sti_is(i);  % Property number
            fem_dat_be(femc_be,4) = mesh; % Mesh fineness
        end

        % Breaking inner top skin where it meets decksection girder,
        % for easier meshing
        if i == length(ss_nst) &&...
                (j)*ss_H(i)/(ss_nst(i)+1) >= ss_H(i)-550 && lb == 0

            lb = 1; % Store breakage
            % Consturct points and line for breaking
            fprintf(fileID,'asm_const_grid_xyz( "%d",@\n',3001);
            fprintf(fileID,'"[%d %d %d]", "Coord 0",  @\n'...
                ,i_xof,0,zof+ss_H(i)-550);
            fprintf(fileID,'asm_create_grid_xyz_created_ids )\n');
            fprintf(fileID,'asm_const_grid_xyz( "%d",@\n',3002);
            fprintf(fileID,'"[%d %d %d]", "Coord 0",  @\n'...
                i_xof+i_l,0,zof+ss_H(i)-550);
            fprintf(fileID,'asm_create_grid_xyz_created_ids )\n');
            fprintf(fileID,...
                'STRING asm_line_2point_created_ids[VIRTUAL]\n');
```

```matlab
        fprintf(fileID,'asm_const_line_2point( "3001",@\n');
        fprintf(fileID,...
            '"point 3002", "point 3001", 0, "", 50., 1,  @\n');
        fprintf(fileID,'asm_line_2point_created_ids )\n');
        % Break surface
        sc = sc+1;
        fprintf(fileID,...
            'STRING sgm_surface_break_c_created_ids[VIRTUAL]\n');
        fprintf(fileID,'sgm_edit_surface_break_v1( "%d",@\n',sc);
        fprintf(fileID,'"Surface %d", FALSE, 3, 0, 0.,@\n',sc-1);
        fprintf(fileID,'"", "", "Curve 3001", @\n');
        fprintf(fileID,'sgm_surface_break_c_created_ids )\n');
        fprintf(fileID,...
            'STRING asm_delete_surface_deleted_ids[VIRTUAL])\n');
        % Delete old surface
        fprintf(fileID,'asm_delete_surface(@\n');
        fprintf(fileID,'"Surface %d", @\n',sc-1);
        fprintf(fileID,'asm_delete_surface_deleted_ids )\n');
        % Renumber new surface
        fprintf(fileID,...
            'STRING sgm_renum_surface_new_ids[VIRTUAL]\n');
        fprintf(fileID,'sgm_renumber( 1, "surface", @\n');
        fprintf(fileID,'"%d", "Surface %d",  @\n',sc-1,sc+1);
        fprintf(fileID,'sgm_renum_surface_new_ids )\n');
        % Store FEM data
        femc_sh = femc_sh + 1;
        fem_dat_sh(femc_sh,1) =  sc;
        fem_dat_sh(femc_sh,2) = ss_plt_is(i);
        fem_dat_sh(femc_sh,3) = mesh;
        % Deletion of break curves
        fprintf(fileID,...
            'STRING asm_delete_curve_deleted_ids[VIRTUAL]\n');
        fprintf(fileID,'asm_delete_curve( "Curve 3001",@\n');
        fprintf(fileID,'asm_delete_curve_deleted_ids )\n');
        % Deletion of break points
        fprintf(fileID,...
            'STRING asm_delete_point_deleted_ids[VIRTUAL]\n');
        fprintf(fileID,'asm_delete_point( "Point 2001:2002",@\n');
        fprintf(fileID,'asm_delete_point_deleted_ids )\n');
        % Store boundary conditions if applicable
        if section == 1 && b == 1
            bcc = bcc + 1;
            bc(bcc,:) = [sc 4 1];
            bcc = bcc + 1;
            bc(bcc,:) = [sc-1 4 1];
        elseif section == 3 && b == 3
            bcc = bcc + 1;
            bc(bcc,:) = [sc 2 1];
            bcc = bcc + 1;
            bc(bcc,:) = [sc-1 2 1];
        end
    end

    % Extrude outer skin
    sc = sc + 1;
    fprintf(fileID,...
        'STRING sgm_sweep_surface_e_created_ids[VIRTUAL]\n');
    fprintf(fileID,'sgm_const_surface_extrude( @\n');
    fprintf(fileID,'"%d", "<%d 0 0>",@\n',sc,i_l);
    fprintf(fileID,'1., 0., "[0 0 0]", "Coord 0",@\n');
    if lb == 0 || lb == 2
        fprintf(fileID,'"Surface %d.4", @\n',sc-3);
        fprintf(fileID,'sgm_sweep_surface_e_created_ids )\n');
    else
        fprintf(fileID,'"Surface %d.4", @\n',sc-4);
        fprintf(fileID,'sgm_sweep_surface_e_created_ids )\n');
        lb = 2;
    end
```

```matlab
% Store FEM and BC data
femc_sh = femc_sh + 1;
fem_dat_sh(femc_sh,1) = sc;
fem_dat_sh(femc_sh,2) = ss_plt_os(i);
fem_dat_sh(femc_sh,3) = mesh;
if section == 1 && b == 1
    bcc = bcc + 1;
    bc(bcc,1) = sc;
    bc(bcc,2) = 4;
    bc(bcc,3) = 1;
elseif section == 3 && b == 3
    bcc = bcc + 1;
    bc(bcc,1) = sc;
    bc(bcc,2) = 2;
    bc(bcc,3) = 1;
end
if j~=1
    femc_be = femc_be +1;
    fem_dat_be(femc_be,1) = sc; % Surface number
    fem_dat_be(femc_be,2) = 1;  % Curve side number
    fem_dat_be(femc_be,3) = ss_sti_os(i);  % Property number
    fem_dat_be(femc_be,4) = mesh;
end

% Extrude stringers
if j == 1
    smesh = mesh;
    if topmesh == 2
        smesh = topmesh;
    end
    sc = sc + 1;
    fprintf(fileID,'sgm_const_surface_extrude( @\n');
    fprintf(fileID,'"%d", "<0 %d 0>", 1., 0.,@\n',sc,g_wss);
    fprintf(fileID,'"[0 0 0]", "Coord 0",  @\n');
    fprintf(fileID,'"Surface %d.1", @\n',sc-1);
    fprintf(fileID,'sgm_sweep_surface_e_created_ids )\n');
    if b == 2
        femc_sh = femc_sh + 1;
        fem_dat_sh(femc_sh,:) =  [sc ss_plt_str(2,i) smesh];
    else
        femc_sh = femc_sh + 1;
        fem_dat_sh(femc_sh,:) =  [sc ss_plt_str(1,i) smesh];
    end
    % Breake stringers and tank top for stiffeners
    coord =  [  i_xof+i_l, (g_wss-650)/2, zof;
        i_xof+i_l,g_wss-(g_wss-650)/2 ,        zof
        i_xof, (g_wss-650)/2, zof;
        i_xof, g_wss-(g_wss-650)/2,        zof];
    sc = prep_string(fileID,coord,sc,2,sc);
    % Store FEM and BC data where applicable
    femc_be = femc_be +1;
    fem_dat_be(femc_be,:) = [sc 2 ss_sti_str(i) smesh];
    femc_be = femc_be +1;
    fem_dat_be(femc_be,:) = [sc 4 ss_sti_str(i) smesh];
    if b == 2
        femc_sh = femc_sh + 1;
        fem_dat_sh(femc_sh,:) =  [sc ss_plt_str(2,i) smesh];
        femc_sh = femc_sh + 1;
        fem_dat_sh(femc_sh,:) =  [sc-1 ss_plt_str(2,i) smesh];
    else
        femc_sh = femc_sh + 1;
        fem_dat_sh(femc_sh,:) =  [sc ss_plt_str(1,i) smesh];
        femc_sh = femc_sh + 1;
        fem_dat_sh(femc_sh,:) =  [sc-1 ss_plt_str(1,i) smesh];
    end
    if section == 1 && b == 1
        bcc = bcc + 1;
        bc(bcc,:) = [sc 3 1];
```

```matlab
                        bcc = bcc + 1;
                        bc(bcc,:) = [sc-1 1 1];
                        bcc = bcc + 1;
                        bc(bcc,:) = [sc-2 1 1];
                    elseif section == 3 && b == 3
                        bcc = bcc + 1;
                        bc(bcc,:) = [sc 1 1];
                        bcc = bcc + 1;
                        bc(bcc,:) = [sc-1 3 1];
                        bcc = bcc + 1;
                        bc(bcc,:) = [sc-2 3 1];
                    end
                end
            end
        end

        % Extrude deck on sidesection
        sc = sc + 1;
        fprintf(fileID,'sgm_const_surface_extrude( "%d", "<0 %d @\n',sc,g_wss);
        fprintf(fileID,'0>", 1., 0., "[0 0 0]", "Coord 0",  @\n');
        fprintf(fileID,...
            '"Surface %d.3", sgm_sweep_surface_e_created_ids )\n',sc-1);
        zof = sum(ss_H);
        femc_sh = femc_sh + 1;
        fem_dat_sh(femc_sh,:) = [sc ss_plt_str(1,length(ss_plt_str)) 1];
        % Break deck for easier meshing
        coord = [  i_xof+i_l, (g_wss-650)/2, zof;
            i_xof+i_l,g_wss-(g_wss-650)/2 ,       zof
            i_xof, (g_wss-650)/2, zof;
            i_xof, g_wss-(g_wss-650)/2,        zof];
        sc = prep_string(fileID,coord,sc,2,sc);
        femc_sh = femc_sh + 1;
        fem_dat_sh(femc_sh,:) = [sc ss_plt_str(1,length(ss_plt_str)) 1];
        femc_sh = femc_sh + 1;
        fem_dat_sh(femc_sh,:) = [sc-1 ss_plt_str(1,length(ss_plt_str)) 1];
        if section == 1 && b == 1
            bcc = bcc + 1;
            bc(bcc,:) = [sc 3 1];
            bcc = bcc + 1;
            bc(bcc,:) = [sc-1 1 1];
            bcc = bcc + 1;
            bc(bcc,:) = [sc-2 1 1];
        elseif section == 3 && b == 3
            bcc = bcc + 1;
            bc(bcc,:) = [sc 1 1];
            bcc = bcc + 1;
            bc(bcc,:) = [sc-1 3 1];
            bcc = bcc + 1;
            bc(bcc,:) = [sc-2 3 1];
        end
        % Delete points and curves used
        fprintf(fileID,'STRING asm_delete_point_deleted_ids[VIRTUAL]\n');
        fprintf(fileID,...
         'asm_delete_point( "Point 1:4000", asm_delete_any_deleted_ids )\n');
        fprintf(fileID,...
         'STRING asm_delete_curve_deleted_ids[VIRTUAL]\n');
        fprintf(fileID,...
         'asm_delete_curve( "Curve 1:1000", asm_delete_curve_deleted_ids )\n');
    end
end
```

double_bottom.m

```matlab
function [sc,fem_dat_sh,fem_dat_be,bc] = double_bottom(fileID,xof,g_lbwf,...
    g_Htt,db_ns,db_W,g_wss,sc,wmh,section,db_plt_os,db_plt_tt,...
    db_plt_wf,db_sti_wf,db_plt_str,db_sti_str,db_sti_os,db_sti_tt,...
    fem_dat_sh,fem_dat_be,db_plt_wf_s,bc)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Function for generation of PATRAN session file commands which will   %
% generate a double bottom section when run.                          %
%                                                                     %
% Written by Halvor L. Aga as part of the work concerning             %
% "Assessement if structural requirements related to LNG fuel tanks." %
% Deadline 10. June 2013                                              %
% Has been made by modification of recorded .ses commands in PATRAN.  %
%                                                                     %
% Input:                                                              %
% fileID - file identification number of the file for writing         %
% xoff - Offset in x direction for the section                       %
% g_lbwf - Length between the web frames                             %
% g_Htt  - Height of the tank top                                    %
% db_ns - number of stiffeners in each distance between the longitudinal %
%          girders                                                   %
% db_W - Width between each of the longitudinal girders              %
% g_wss - Width of the side section                                  %
% sc - Counter for number of created surfaces                        %
% wmh - Width of a manhole                                           %
% section - Information of placement of the section                  %
% db_plt_os - property numbers for plating in outer skin             %
% db_plt_tt - property numbers for plating in tank top               %
% db_plt_wf - property numbers for plating in web frame              %
% db_plt_str- property numbers for plating in stringers              %
% db_sti_str- property numbers for stiffeners in stringers           %
% db_sti_os - property numbers for stiffeners in outer skin          %
% db_sti_tt - property numbers for stiffeners in tank top            %
% fem_dat_sh - matrix containing data regarding shell elements       %
% fem_dat_be - matrix containing data regarding beam elements        %
% db_plt_wf_s - property numbers for plating in web frame where holes or %
%              manholes                                              %
%                                                                     %
% bc - matrix containing data regarding boundary conditions          %
%                                                                     %
% Output:                                                             %
% sc - Counter for number of created surfaces                        %
% fem_dat_sh - matrix containing data regarding shell elements       %
% fem_dat_be - matrix containing data regarding beam elements        %
% bc - matrix containing data regarding boundary conditions          %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Determination of femc, finite elements counter
if (size(fem_dat_sh,2)==1); femc_sh = 0;
else femc_sh = size(fem_dat_sh,1);
end
% Determination of femc_be, finite elements counter for beam elements
if (size(fem_dat_be,2)==1); femc_be = 0;
else femc_be = size(fem_dat_be,1);
end
% Determination of bcc, counter for boundary conditions
if (size(bc,2)==1); bcc = 0;
else bcc = size(bc,1);
end




xofm = [xof, xof+(g_lbwf-800)/2, xof+(g_lbwf+800)/2];
lbwfm = [(g_lbwf-800)/2, 800, (g_lbwf-800)/2];


for b = 1:3
```

```matlab
    i_xof = xofm(b);
    i_l = lbwfm(b);
    pc = 0; % Point counter
    cc = 0; % Curve counter
    sco = sc ; % Store number of surfaces generated
    %-------------------------------------------------------------------------
    % Point generation

    % Origo
    fprintf(fileID,'STRING asm_create_grid_xyz_created_ids[VIRTUAL]\n');
    pc = pc + 1;
    fprintf(fileID,'asm_const_grid_xyz( "%d", "[%d 0 0]", "Coord 0",
@\n',pc,i_xof);
    fprintf(fileID,'asm_create_grid_xyz_created_ids )\n');
    % Arc
    % Calculate position of points in arc
    if g_wss <= g_Htt;
        y = 0;
        z = g_wss-g_Htt;
    else
        y = g_Htt;
        z = -g_Htt;
    end
    pc = pc + 1;
    fprintf(fileID,'asm_const_grid_xyz( "%d", "[%d %d %d]",  @\n',pc,i_xof,y,z);
    fprintf(fileID,'"Coord 0", asm_create_grid_xyz_created_ids )\n');
    pc = pc + 1;
    fprintf(fileID,'asm_const_grid_xyz( "%d", "[%d %d %d]",  @\n',pc,i_xof,g_wss,-
g_Htt);
    fprintf(fileID,'"Coord 0", asm_create_grid_xyz_created_ids )\n');
    % Double hull structure
    % Along stringers on tank top
    % Calculation of y values for first point
    y = g_wss;
    pc = pc+1;
    fprintf(fileID,'asm_const_grid_xyz( "%d", "[%d %d 0]",  @\n',pc,i_xof,y);
    fprintf(fileID,'"Coord 0", asm_create_grid_xyz_created_ids )\n');
    for i = 1:length(db_ns)
        for j = 1:db_ns(i)+1
            % Calculating point numbers and position
            y = y + db_W(i)/(db_ns(i)+1);
            pc = pc + 1;
            % Printing necessary commands for generation of point i
            fprintf(fileID,'asm_const_grid_xyz( "%d", "[%d %d 0]",
@\n',pc,i_xof,y);
            fprintf(fileID,'"Coord 0", asm_create_grid_xyz_created_ids )\n');
        end
    end

    % Point generation finished
    %-------------------------------------------------------------------------
----
    % Curve generation

    % Radius
    R = min(g_Htt,g_wss);
    if g_Htt<=g_wss
        cc = cc + 1;
        fprintf(fileID,'STRING sgm_create_curve_2d_created_ids[VIRTUAL]\n');
        fprintf(fileID,'sgm_const_curve_2d_arc2point_v2( "%d", 2, %d, FALSE, FALSE,
1, "Coord 0.1",  @\n',cc,R);
        fprintf(fileID,'"", "point 1", "point 2", FALSE,
sgm_create_curve_2d_created_ids )\n');
        cc = cc+1;
        fprintf(fileID,'STRING asm_line_2point_created_ids[VIRTUAL]\n');
        fprintf(fileID,'asm_const_line_2point( "%d", "point 2", "point 3", 0, "",
50., 1,  @\n',cc);
        fprintf(fileID,'asm_line_2point_created_ids )\n');
```

```matlab
    else
        cc = cc + 1;
        fprintf(fileID,'STRING sgm_create_curve_2d_created_ids[VIRTUAL]\n');
        fprintf(fileID,'sgm_const_curve_2d_arc2point_v2( "1", 2, %d, FALSE, FALSE,
1, "Coord 0.1",  @\n',R);
        fprintf(fileID,'"", "point 2", "point 3", FALSE,
sgm_create_curve_2d_created_ids )\n');
        cc = cc+1;
        fprintf(fileID,'STRING asm_line_2point_created_ids[VIRTUAL]\n');
        fprintf(fileID,'asm_const_line_2point( "%d", "point 1", "point 2", 0, "",
50., 1,  @\n',cc);
        fprintf(fileID,'asm_line_2point_created_ids )\n');
    end
    cc = cc+1;
    fprintf(fileID,'asm_const_line_2point( "%d", "point 1", "point 4", 0, "", 50.,
1,  @\n',cc);
    fprintf(fileID,'asm_line_2point_created_ids )\n');
    cc = cc+1;
    fprintf(fileID,'asm_const_line_2point( "%d", "point 3", "point 4", 0, "", 50.,
1,  @\n',cc);
    fprintf(fileID,'asm_line_2point_created_ids )\n');

    % Tank top and outer skin
    for i = 1:sum(db_ns)+length(db_ns)
        cc=cc+1;
        fprintf(fileID,'asm_const_line_2point( "%d", "point %d", "point %d", 0, "",
50., 1,  @\n',cc,i+3,i+4);
        fprintf(fileID,'asm_line_2point_created_ids )\n');
    end

    % Curve generation finished
    %------------------------------------------------------------------------------
----
    % Surface generation

    % Web frame
    fprintf(fileID,'STRING sgm_sweep_surface_e_created_ids[VIRTUAL]\n');
    % Decide which panel is the first with manhole
    firstbrake = db_ns(1)+1+floor((db_ns(2)+1)/2)-
ceil(wmh/2/(db_W(2)/(db_ns(2)+1)))+1+sco;
    % Decide which panel is the last with manhole
    lastbrake  =
db_ns(1)+1+floor((db_ns(2)+2)/2)+ceil(wmh/2/(db_W(2)/(db_ns(2)+1)))+sco;
    % Between first and last longitudinal girder
    count = 4;
    for i = 1:length(db_ns)
        for j = 1:db_ns(i)+1
            sc = sc+1;
            count = count + 1;
            fprintf(fileID,'sgm_const_surface_extrude( "%d", "<0 0 %d>", 1., 0.,
"[0 0 0]", "Coord 0", @\n',sc,-g_Htt);
            fprintf(fileID,' "Curve %d", sgm_sweep_surface_e_created_ids
)\n',count);
            % Breaking for easier meshing
            coord = [i_xof, g_wss+sum(db_W(1:i-1))+(j-1)*db_W(i)/(db_ns(i)+1), -
(g_Htt/2-325);
                    i_xof, g_wss+sum(db_W(1:i-1))+(j-1)*db_W(i)/(db_ns(i)+1), -
(g_Htt/2+325) ;
                    i_xof, g_wss+sum(db_W(1:i-1))+j*db_W(i)/(db_ns(i)+1), -(g_Htt/2-
325) ;
                    i_xof, g_wss+sum(db_W(1:i-1))+j*db_W(i)/(db_ns(i)+1), -
(g_Htt/2+325) ];
            sc = prep_string(fileID,coord,sc,1,sc);

            % Store finite element data if not the first section
            % Vary properties
            for k = 1:3
```

```matlab
                        if section ~=1 && b == 1 && sc<=3*lastbrake+2-2*sco &&
sc>=3*firstbrake-2*sco
                            femc_sh = femc_sh + 1;
                            fem_dat_sh(femc_sh,1) = sc-k+1; % Surface number
                            fem_dat_sh(femc_sh,2) = db_plt_wf_s(1); % Property number
                            fem_dat_sh(femc_sh,3) = 1; % Mesh fineness
                            if j~=1
                                femc_be = femc_be +1;
                                fem_dat_be(femc_be,1) = sc-k+1; % Surface number
                                fem_dat_be(femc_be,2) = 1;  % Curve side number
                                fem_dat_be(femc_be,3) = db_sti_wf(i);  % Property number
                                fem_dat_be(femc_be,4) = 1; % Mesh fineness
                            end
                        elseif  section ~=1 && b == 1 && (sc ==
sco+sum(db_ns)+length(db_ns)-ceil(db_ns(length(db_ns))/2)...
                            ||sc == sco+sum(db_ns)+length(db_ns)-
ceil(db_ns(length(db_ns))/2)+1);
                            femc_sh = femc_sh + 1;
                            fem_dat_sh(femc_sh,1) = sc-k+1;
                            fem_dat_sh(femc_sh,2) = db_plt_wf_s(2);
                            fem_dat_sh(femc_sh,3) = 1;
                            if j~=1
                                femc_be = femc_be +1;
                                fem_dat_be(femc_be,1) = sc-k+1;
                                fem_dat_be(femc_be,2) = 1;
                                fem_dat_be(femc_be,3) = db_sti_wf(i);
                                fem_dat_be(femc_be,4) = 1;
                            end
                        elseif section ~=1 && b == 1
                            femc_sh = femc_sh + 1;
                            fem_dat_sh(femc_sh,1) = sc-k+1;
                            fem_dat_sh(femc_sh,2) = db_plt_wf(i+1);
                            fem_dat_sh(femc_sh,3) = 1;
                            if j~=1
                                femc_be = femc_be +1;
                                fem_dat_be(femc_be,1) = sc-k+1;
                                fem_dat_be(femc_be,2) = 1;
                                fem_dat_be(femc_be,3) = db_sti_wf(i);
                                fem_dat_be(femc_be,4) = 1;
                            end
                        end
                    end
                end
        end

        % Webframe inside radius
        sc = sc + 1;
        fprintf(fileID,'STRING sgm_surface_4edge_created_ids[VIRTUAL]\n');
        fprintf(fileID,'sgm_const_surface_4edge( "%d", "Curve 1", "Curve 2", "Curve 3",
@\n',sc);
        fprintf(fileID,'"Curve 4", sgm_surface_4edge_created_ids )\n');
        % Store finite element data if not the first section
        if section ~=1 && b == 1
            femc_sh = femc_sh + 1;
            fem_dat_sh(femc_sh,:) = [sc db_plt_wf(1) 1];
        end

        % Extrude section to surfaces from curves
        fprintf(fileID,'STRING sgm_sweep_surface_e_created_ids[VIRTUAL]\n');

        % Radius
        for i = 1:2
            sc = sc+1;
            fprintf(fileID,'sgm_const_surface_extrude( "%d", "<%d 0 0>", 1., 0., "[0 0
0]", "Coord 0", @\n',sc,i_l);
            fprintf(fileID,' "Curve %d", sgm_sweep_surface_e_created_ids )\n',i);
            % Store finite element data
            femc_sh = femc_sh + 1;
```

```matlab
        fem_dat_sh(femc_sh,1) = sc;
        fem_dat_sh(femc_sh,2) =    db_plt_os(1) ;
         if  section == 2
            fem_dat_sh(femc_sh,3) = 2;
         else
             fem_dat_sh(femc_sh,3) = 1;
         end
        % Store boundary conditions if first or last section
        if section == 1 && b == 1
            bcc = bcc + 1;
            bc(bcc,1) = sc; % Surface number
            bc(bcc,2) = 4;  % Side number
            bc(bcc,3) = 1;  % Boundary condition type
        elseif section == 3 && b == 3
            bcc = bcc + 1;
            bc(bcc,1) = sc;
            bc(bcc,2) = 2;
            bc(bcc,3) = 1;
        end
    end

    % Tanktop
    count = 4;
    for i = 1:length(db_ns)
        for j = 1:db_ns(i)+1
            count = count + 1;
            sc = sc+1;
            fprintf(fileID,'sgm_const_surface_extrude( "%d", "<%d 0 0>", 1., 0.,
"[0 0 0]", "Coord 0", @\n',sc,i_l);
            fprintf(fileID,' "Curve %d", sgm_sweep_surface_e_created_ids
)\n',count);
            femc_sh = femc_sh + 1;
            fem_dat_sh(femc_sh,:) = [sc db_plt_tt(i) 1];
            if j ~= 1
                femc_be = femc_be +1;
                fem_dat_be(femc_be,:) = [sc 1 db_sti_tt(i) 1];
            end
            if section == 1 && b == 1
                bcc = bcc + 1;
                bc(bcc,:) = [sc 4 1];
            elseif section == 3 && b == 3
                bcc = bcc + 1;
                bc(bcc,:) = [sc 2 1];
            end
        end
    end

    % Outer skin
    count = sco-2;
    for i = 1:length(db_ns)
        for j = 1:db_ns(i)+1
            count = count + 3;
            sc = sc + 1;
            fprintf(fileID,'sgm_const_surface_extrude( "%d", "<%d 0 0>", 1., 0.,
"[0 0 0]", "Coord 0", @\n',sc,i_l);
            fprintf(fileID,' "Surface %d.2", sgm_sweep_surface_e_created_ids
)\n',count);
            femc_sh = femc_sh + 1;
            fem_dat_sh(femc_sh,:) = [sc db_plt_os(i+1) 1];
            if j ~= 1
                femc_be = femc_be +1;
                fem_dat_be(femc_be,:) = [sc,1,db_sti_os(1) 1];
            end
            if section == 1 && b == 1
                bcc = bcc + 1;
                bc(bcc,:) = [sc 4 1];
            elseif section == 3 && b == 3
                bcc = bcc + 1;
```

```matlab
                    bc(bcc,:) = [sc 2 1];
                end
            end
        end

        % Longitudinal girders
        count = sco + 1;
        for i = 1:length(db_ns)+1
            for k = 1:3
                sc = sc + 1;
                if i ~= 1 && k==1; count = count + 3*db_ns(i-1)+3;  end
                if i ~= length(db_ns)+1
                    fprintf(fileID,'sgm_const_surface_extrude( "%d", "<%d 0 0>", 1.,
0., "[0 0 0]", "Coord 0", @\n',sc,i_l);
                    fprintf(fileID,' "Surface %d.1", sgm_sweep_surface_e_created_ids
)\n',count+k-1);
                    if k==3
                        femc_be = femc_be + 1;
                        fem_dat_be(femc_be,:) = [sc-1 3 db_sti_str(i) 1];
                        femc_be = femc_be + 1;
                        fem_dat_be(femc_be,:) = [sc-2 1 db_sti_str(i) 1];
                        femc_sh = femc_sh + 1;
                        fem_dat_sh(femc_sh,:) = [sc db_plt_str(2,i) 1];
                    else
                        femc_sh = femc_sh + 1;
                        fem_dat_sh(femc_sh,:) = [sc db_plt_str(1,i) 1];
                    end
                    if section == 1 && b == 1
                        bcc = bcc + 1;
                        bc(bcc,:) = [sc 4 1];
                    elseif section == 3 && b == 3
                        bcc = bcc + 1;
                        bc(bcc,:) = [sc 2 1];
                    end
                else
                    fprintf(fileID,'sgm_const_surface_extrude( "%d", "<%d 0 0>", 1.,
0., "[0 0 0]", "Coord 0", @\n',sc,i_l);
                    fprintf(fileID,' "Surface %d.3", sgm_sweep_surface_e_created_ids
)\n',count-1+1-k);
                    femc_sh = femc_sh + 1;
                    fem_dat_sh(femc_sh,:) = [sc db_plt_str(1,i) 1];
                    if section == 1 && b == 1
                        bcc = bcc + 1;
                        bc(bcc,:) = [sc 4 1];
                    elseif section == 3 && b == 3
                        bcc = bcc + 1;
                        bc(bcc,:) = [sc 2 1];
                    end
                end
            end
        end

        % Surface generation finished
        %----------------------------------------------------------------------------
----
        % Cleaning

        % Delete all points and curves used
        fprintf(fileID,'STRING asm_delete_point_deleted_ids[VIRTUAL]\n');
        fprintf(fileID,'asm_delete_point( "Point 1:2050", asm_delete_any_deleted_ids
)\n');
        fprintf(fileID,'STRING asm_delete_curve_deleted_ids[VIRTUAL]\n');
        fprintf(fileID,'asm_delete_curve( "Curve 1:2050", asm_delete_curve_deleted_ids
)\n');

end
end
```

mesh.m

```matlab
function [offsetm,bpname,finemesh] = mesh(fileID,fem_dat_sh,fem_dat_be,...
    bc,filesID)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Function for generation of PATRAN session file commands which will      %
% generate neccessary commands for creating materials, sections and       %
% meshing in a ses script.                                                %
%                                                                         %
% Written by Halvor L. Aga as part of the work concerning                 %
% "Assessement if structural requirements related to LNG fuel tanks."     %
% Deadline 10. June 2013                                                   %
% Has been made by modification of recorded .ses commands in PATRAN.      %
%                                                                         %
% Build-up:                                                               %
%    1. Create sections                                                   %
%    2. Generate mesh for stiffeners                                      %
%    3. Sew surfaces                                                      %
%    4. Generate mesh for surfaces                                        %
%    5. Equivalence nodes                                                 %
%                                                                         %
% Input:                                                                  %
% fileID - file identification number of the file for writing            %
% fem_dat_sh - matrix containing data regarding shell elements            %
% fem_dat_be - matrix containing data regarding beam elements             %
% bc - matrix containing data regarding boundary conditions               %
% filesID - Vector containing fileID of the open files that is used       %
%           througout the program                                         %
%                                                                         %
% Output:                                                                 %
% offsetm - matrix containing data for beams about offset                 %
% bpname  - vector containing names of the beams in offsetm               %
% finemesh - Mesh size for fine meshing                                   %
%                                                                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Nonlinear steel
fprintf(fileID,'material.create( "Analysis code ID", 10001,@\n');
fprintf(fileID,'"Analysis type ID", 1, "nlSteel",@\n');
fprintf(fileID,...
    '0, "Date: 06-Mar-13         Time: 20:42:28", "Isotropic", 1,@\n');
fprintf(fileID,'"Directionality", 1, "Linearity", 3, "Homogeneous",@\n');
fprintf(fileID,'0, "Elastoplastic", 3, "Model Options & IDs",@\n');
fprintf(fileID,'["Piecewise Linear(MAT24)", "Bilinear",@\n');
fprintf(fileID,'"N/A.", "Cowper Symonds", ""], [11026, 11028,@\n');
fprintf(fileID,'11022, 11031, 0], "Active Flag", 1, "Create",@\n');
fprintf(fileID,'10, "External Flag", FALSE, "Property IDs",@\n');
fprintf(fileID,'["Density", "Elastic Modulus", "Poisson Ratio",@\n');
fprintf(fileID,'"Yield Stress"], [16, 2, 5, 1011, 0],@\n');
fprintf(fileID,...
    '"Property Values", ["7.85e-9", "2.1e5", "0.3", "345", ""] )\n');

% Create sections
sectionID = filesID(3);
for i = 1:4 % Omitting lines
    fgets(sectionID);
end

% Retrieve number of shell sections
n=strread(fgets(sectionID));
fgets(sectionID); % omit line
% Shell sections
for i = 1:n
    %[section Name Materaltype Thickness]
    [a b c d] = strread(fgets(sectionID),'%s %s %s %f');
    fprintf(fileID,...
        'elementprops_create( "shell_%s_%s",@\n',char(a),char(b));
    fprintf(fileID,'51, 25, 35, 11004, 1, 20,@\n');
    fprintf(fileID,...
        '[13, 20, 36, 1004, 11044, 11136, 11027, 1011, 11182],@\n');
    fprintf(fileID,'[5, 2, 1, 1, 4, 3, 4, 1, 1],@\n');
    fprintf(fileID,...
        '["m:%s", "", "%f", "0.833", "", "5", "", "", ""],@\n',char(c),d);
    fprintf(fileID,'"" )\n');
    % Store property name
    v1 = {'shell_'};
    v2 = char(a);
```

```matlab
        v3 = {'_'};
        v4 = char(b);
        spname(i) = strcat(v1,v2,v3,v4);
    end

    for i = 1:2 % Omit lines
        fgets(sectionID);
    end

    % Retrieve number of beam sections
    m=strread(fgets(sectionID));
    fgets(sectionID); % omitting line
    offsetm = zeros(m,3);
    % Beam sections
    for i = 1:m
        % [section Name Materaltype Thickness Height Directionvector(xyz)]
        [a b c d e f g h m n o] = ...
            strread(fgets(sectionID),'%s %s %s %f %f %f %f %f %f %f %f');
        fprintf(fileID,'beam_section_create( @\n');
        fprintf(fileID,...
            '"FB_%3.1fx%3.1f_%s_%s", "BAR", ["%3.1f", "%3.1f"] )\n'...
            ,e,d,char(a),char(b),d,e);
        fprintf(fileID,...
            'elementprops_create( "beam_%s_%s",@\n',char(a),char(b));
        fprintf(fileID,'11, 38, 50, 11003, 1, 20, [39, 13, 6, 1004,@\n');
        fprintf(fileID,'11022, 11139, 11140, 1011, 11182], [11, 5, 2, 1, @\n');
        fprintf(fileID,...
            '4, 4, 4, 1, 1], ["FB_%3.1fx%3.1f_%s_%s", "m:%s",@\n'...
            ,e,d,char(a),char(b),char(c));
        fprintf(fileID,...
            '"<%f %f %f>", "0.833", "", "Center", "t=+1", "", ""], "" )\n',f,g,h);
        % Storing property name
        v1 = {'beam_'};
        v2 = char(a);
        v3 = {'_'};
        v4 = char(b);
        bpname(i,:) = strcat(v1,v2,v3,v4);
        % Store offset data
        offsetm(i,:) = [m n o];
    end

    %Generate mesh for stiffeners
    %Collapse fem_dat_be
    fem_dat_be_coll = collapse(fem_dat_be);
    fprintf(fileID,'INTEGER fem_create_mesh_curve_num_nodes\n');
    fprintf(fileID,'INTEGER fem_create_mesh_curve_num_elems\n');
    fprintf(fileID,'STRING fem_create_mesh_c_nodes_created[VIRTUAL]\n');
    fprintf(fileID,'STRING fem_create_mesh_c_elems_created[VIRTUAL]\n');
    fgets(sectionID); % Omit line
    % Read data for mesh fineness
    [coursemesh finemesh]=strread(fgets(sectionID));
    % Write commands for mesh on surface edges
    for i = 1:size(fem_dat_be_coll,1)
        if fem_dat_be_coll(i,5) == 1;
            esize = coursemesh;
        elseif fem_dat_be_coll(i,5) == 2;
            esize = finemesh;
        end
        fprintf(fileID,...
            'fem_create_mesh_curv_1( "Surface %d:%d.%d", 16384, %d.,@\n'...
            ,fem_dat_be_coll(i,1),fem_dat_be_coll(i,2),...
            fem_dat_be_coll(i,3),esize);
        fprintf(fileID,'"Bar2", "#", "#", "Coord 0", "Coord 0",@\n');
        fprintf(fileID,'fem_create_mesh_curve_num_nodes,@\n');
        fprintf(fileID,'fem_create_mesh_curve_num_elems,@\n');
        fprintf(fileID,'fem_create_mesh_c_nodes_created,@\n');
        fprintf(fileID,'fem_create_mesh_c_elems_created )\n');
        fprintf(fileID,'fem_associate_elems_to_ep( "%s",@\n',...
            char(bpname(fem_dat_be_coll(i,4))));
        fprintf(fileID,'fem_create_mesh_c_elems_created,@\n');
        fprintf(fileID,'fem_create_mesh_curve_num_elems )\n');
    end

    % Collapse boundary condition matrix
    bc = collapse(bc);
    % Create boundary condition sets
    for i = 1:size(bc,1)
```

```matlab
            fprintf(fileID,'loadsbcs_create2( "%d_bc%d", "Displacement",@\n',...
                bc(i,4),i);
            fprintf(fileID,'"Nodal", "", "Static", ["Surface %d:%d.%d"],@\n',...
                bc(i,1),bc(i,2),bc(i,3));
            fprintf(fileID,'"Geometry", "Coord 0", "1.", ["<0 0 0>", @\n');
            fprintf(fileID,'"", "", ""],@\n');
            fprintf(fileID,' ["", "", "", ""] )\n');
end

% Generate mesh for shell structure
fprintf(fileID,'INTEGER fem_create_mesh_surfa_num_nodes\n');
fprintf(fileID,'INTEGER fem_create_mesh_surfa_num_elems\n');
fprintf(fileID,'STRING fem_create_mesh_s_nodes_created[VIRTUAL]\n');
fprintf(fileID,'STRING fem_create_mesh_s_elems_created[VIRTUAL]\n');

% Collapse fem_dat_sh
fem_dat_sh_coll=collapse(fem_dat_sh);

% Sew surfaces
fprintf(fileID,'STRING sgm_edit_surface_se_surface_ids[VIRTUAL]\n');
fprintf(fileID,'sgm_edit_surface_sew(@\n');
fprintf(fileID,'"Surface 1:%d", sgm_edit_surface_se_surface_ids )\n'...
    ,fem_dat_sh_coll(size(fem_dat_sh_coll,1),2));

% Create shell mesh
for i = 1:size(fem_dat_sh_coll,1)
    if fem_dat_sh_coll(i,4) == 1;
        esize = coursemesh;
    elseif fem_dat_sh_coll(i,4) == 2;
        esize = finemesh;
    end
    fprintf(fileID,'fem_create_mesh_surf_4( "Hybrid", 49664,@\n');
    fprintf(fileID,...
        '"Surface %d:%d", 4, ["%d", "0.1", "0.2", "1.0"],@\n',...
        fem_dat_sh_coll(i,1),fem_dat_sh_coll(i,2),esize);
    fprintf(fileID,'"Quad4", "#", "#", "Coord 0",@\n');
    fprintf(fileID,'fem_create_mesh_surfa_num_nodes,@\n');
    fprintf(fileID,'fem_create_mesh_surfa_num_elems,@\n');
    fprintf(fileID,'fem_create_mesh_s_nodes_created,@\n');
    fprintf(fileID,'fem_create_mesh_s_elems_created )\n');
    fprintf(fileID,'fem_associate_elems_to_ep( "%s",@\n',...
        char(spname(fem_dat_sh_coll(i,3))));
    fprintf(fileID,'fem_create_mesh_s_elems_created,@\n');
    fprintf(fileID,'fem_create_mesh_surfa_num_elems )\n');
end

%Equivalence nodes
fprintf(fileID,'REAL fem_equiv_all_x_equivtol_ab\n');
fprintf(fileID,'INTEGER fem_equiv_all_x_segment\n');
fprintf(fileID,'fem_equiv_all_group4( [" "], 0, "", @\n');
fprintf(fileID,'1, 1, 10., FALSE, fem_equiv_all_x_equivtol_ab, @\n');
fprintf(fileID,'fem_equiv_all_x_segment )\n');
end
```

keyman.m

```matlab
function [] = keyman(filename,addfilename,finishfilename,ofm,bn)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function adds the keyword entries given in a file called         %
% the variable stored in addfilename to a file with name finishfilename %
% and reads nodes, elements, bc's, sections and parts from the file with %
% filename stored as variable in filename.                              %
%                                                                       %
% Written by Halvor L. Aga as part of the work concerning              %
% "Assessement if structural requirements related to LNG fuel tanks."  %
% Deadline 10. June 2013                                                %
% Has been made by modification of recorded .ses commands in PATRAN.   %
%                                                                       %
% Input:                                                                %
% filename      - Path and name of the .ses file containing model data %
%                 from PATRAN                                           %
% addfilename   - Path and name of the file containing keyword commands %
%                 necessary for the analysis                           %
% finishfilename - Desired name of the finished file                   %
% ofm           - Matrix containing data about offsets                  %
% bn            - Corresponding names to the offset matrix             %
%                                                                       %
% Output:                                                               %
% None                                                                  %
%                                                                       %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Open files
keywordID = fopen(filename,'r');
addkeyID = fopen(addfilename,'r');
finishID = fopen(finishfilename,'w');

% Create keyword entries from addfilename file
for i = 1:2 % Omitting lines
    fgetl(addkeyID);
end

n = strread(fgets(addkeyID));
for i = 1:n
    line = fgets(addkeyID);
    fprintf(finishID,line);
end


% Close file with lines added
fclose(addkeyID);

% Adding model data from the file named filename
% Omitting lines in file until *NODE is reached
S = 0;
while strcmp(strcat(line),'*NODE') == 0
    line = fgets(keywordID);
end

S = 0;
% Printing file lines until $ is reached
while strcmp(strcat(line),'$') == 0
    % Special treatment of sections starting with *PART
    if strcmp(strcat(line),'*PART') == 1;
        fprintf(finishID,line);
        line = fgets(keywordID);
        part = line;
        fprintf(finishID,part);
        line = fgets(keywordID);
        fprintf(finishID,line);
        line = fgets(keywordID);
        fprintf(finishID,line);
```

```matlab
        % Changing element type from beam to beam
        % with offset and orientation
        if strcmp(strcat(line),'*SECTION_BEAM');
            line = fgets(keywordID);
            fprintf(finishID,line);
            line = fgets(keywordID);
            [d1 d2 d3 d4 d5 d6] = strread(line);
            fprintf(finishID,...
                '\t\t %.2f %9.2f %9.2f %9.2f %8.2f %9.2f\n'...
                ,d1,d2,d3,d4,d5,-d6);
            storeline = line;
            line = fgets(keywordID);
            fprintf(finishID,'*ELEMENT_BEAM_ORIENTATION_OFFSET');
            fprintf(finishID,'\n');
            for i = 1:length(bn)
                if strcmp(strcat(part),bn(i))== 1
                    c = i;
                end
            end
            line = fgets(keywordID);
            while strcmp(strcat(line),'*PART') == 0 &&...
                    strcmp(strcat(line),'$') == 0
                line2 = fgets(keywordID);
                [b1 b2 b3 b4 b5] = strread(line);
                [c1 c2 c3] = strread(line2);
                [a1 a2 a3 a4 a5 a6] = strread(storeline);
                fprintf(finishID,line);
                fprintf(finishID,line2);
                fprintf(finishID,...
                    '\t\t %.2f %9.2f %9.2f %9.2f %9.2f %9.2f %9.2f',...
                    ofm(c,1),ofm(c,2),ofm(c,3),ofm(c,1),ofm(c,2),ofm(c,3));
                fprintf(finishID,'\n');
                line = fgets(keywordID);
            end
        end
    end
    if strcmp(strcat(line),'*PART') == 0
        fprintf(finishID,line);
        line = fgets(keywordID);
    end
end
end

% Read and omit lines until boundary conditions is reached
line = fgets(keywordID);
while length(line) <= 10||strcmp(strcat(line(1:11)),'$ LBC set :') == 0
    line = fgets(keywordID);
end

% Copy and change the boundary conditions
while(strcmp(strcat(line),'*SET_NODE_LIST_GENERATE') == 0)
    ax = line(13);
    fgets(keywordID); % omitting line
    line = fgets(keywordID);
    fprintf(finishID,line);
    line = fgets(keywordID); % omitting line
    nmbr  = line(1:10);
    bc1 = '          0         1         1         1         1         1
1\n';
    bc2 = '          0         1         1         1         1         1
1\n';
    % Implement the right boundary conditions
    if strcmp(strcat(ax),'1') == 1
        line = strcat(nmbr,bc1);
    elseif strcmp(strcat(ax),'2') == 1
        line = strcat(nmbr,bc2);
    end
    % Copy lines until $ is reached
    while (strcmp(strcat(line(1)),'$') == 0)
```

```matlab
        fprintf(finishID,line);
        line = fgets(keywordID);
    end
    line = fgets(keywordID);
end
% Printing end sign to file
fprintf(finishID,'*END');

% Closing files used
fclose(keywordID);
fclose(finishID);

end
```

postscript.m

```matlab
function [] = postscript(WF,STR,loc,fileID,studyname)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function generates the neccessary session file commands to create  %
% a postscript file for generating a energy indentation curve from four   %
% different bow position simulations                                      %
%                                                                         %
% Written by Halvor L. Aga as part of the work concerning                 %
% "Assessement if structural requirements related to LNG fuel tanks."     %
% Deadline 10. June 2013                                                  %
% Has been made by modification of recorded .ses commands in PATRAN.      %
%                                                                         %
%                                                                         %
% Input:                                                                  %
%       WF - number of webframes in a tank hold                           %
%       STR - number of stringers in sidesection                         %
%       loc - filepath to folder where the results are stored             %
%       fileID - identification key to the postprocessing file            %
%       studyname - name of the study                                     %
%                                                                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

U = (2*STR+1)*(2*WF+1);
W = [STR*WF/U (STR+1)*WF/U (STR+1)*(WF+1)/U STR*(WF+1)/U];

% Preparing the intividual curves
for i = 1:4
fprintf(fileID,'ascii rcforc open "%s\\bow_pos_#%d\\rcforc" 0\n',loc,i);
fprintf(fileID,'ascii rbdout open "%s\\bow_pos_#%d\\rbdout" 0\n',loc,i);
fprintf(fileID,'ascii rcforc plot 2 Sl-2\n');
fprintf(fileID,'xyplot 1 operation inverty all\n');
fprintf(fileID,...
    'xyplot 1 savefile curve_file "%s\\bow_pos_#%d\\Force%d" 1 all\n',loc,i,i);
fprintf(fileID,'ascii rbdout plot 5 1\n');
fprintf(fileID,...
    'xyplot 1 savefile curve_file "%s\\bow_pos_#%d\\Displacement%d" 1 all\n'...
    ,loc,i,i);
fprintf(fileID,'cross Force%d~1 Displacement%d~1 1000\n',i,i);
fprintf(fileID,...
    'xyplot 1 savefile curve_file "%s\\bow_pos_#%d\\ForceDisplacement%d" 1
all\n'...
    ,loc,i,i);
fprintf(fileID,'print jpg "%s\\bow_pos_#%d\\FoDisp%d.jpg" LANDSCAPE nocompress
gamma 1.000 transparent 0x000000 enlisted "PlotWindow-1" \n',loc,i,i);
fprintf(fileID,'xyplot 1 operation integrate all\n');
fprintf(fileID,'xyplot 1 curvelegend 1//1 %s_bowpos#%d\n',studyname,i);
fprintf(fileID,'xyplot 1 legend on\n');
fprintf(fileID,'xyplot 1 legendlabel ""\n');
fprintf(fileID,...
    'xyplot 1 savefile curve_file "%s\\bow_pos_#%d\\EnergyDisplacement%d" 1
all\n'...
    ,loc,i,i);
fprintf(fileID,'print jpg "%s\\bow_pos_#%d\\EnDisp%d.jpg" LANDSCAPE nocompress
gamma 1.000 transparent 0x000000 enlisted "PlotWindow-1" \n',loc,i,i);
fprintf(fileID,'removefile "%s\\bow_pos_#%d\\Force%d"\n',loc,i,i);
fprintf(fileID,'removefile "%s\\bow_pos_#%d\\Displacement%d"\n',loc,i,i);
fprintf(fileID,'removefile "%s\\bow_pos_#%d\\ForceDisplacement%d"\n',loc,i,i);
fprintf(fileID,'removefile "%s\\bow_pos_#%d\\EnergyDisplacement%d"\n',loc,i,i);
fprintf(fileID,'deletewin 1\n');
fprintf(fileID,'ascii rcforc uload\n');
fprintf(fileID,'ascii rbdout uload\n');
end

% Combining to form one energy curve
for i = 1:4
    if i >= 2
    fprintf(fileID,'addplot\n');
    end
```

```matlab
    fprintf(fileID,'open xydata
"%s\\bow_pos_#%d\\EnergyDisplacement%d"\n',loc,i,i);
    fprintf(fileID,'show "EnergyDisplacement%i" 0\n',i);
    fprintf(fileID,'removefile "%s\\bow_pos_#%d\\EnergyDisplacement%d"\n',loc,i,i);
end

% Scaling according to scale factors
for i = 1:4
fprintf(fileID,'xyplot 1 select clear\n');
fprintf(fileID,'xyplot 1 select %d\n',i);
fprintf(fileID,'xyplot 1 xoffset 0 yoffset 0 xscale 1.000000 yscale %f \n',W(i));
end

% Summing the curves

fprintf(fileID,'xyplot 1 operation sum_curves all\n');

% Changing axe titles and gridlines
fprintf(fileID,'xyplot 1 xmin -100 \n');
fprintf(fileID,'xyplot 1 xmax 3680\n');
fprintf(fileID,'xyplot 1 xoffset -100\n');
fprintf(fileID,'xyplot 1 xtitle "Indentation [mm]"\n');
fprintf(fileID,'xyplot 1 ytitle "Energy [J]"\n');
fprintf(fileID,'xyplot 1 curvelegend 1/1 "EnergyIndentation_%s\n"',studyname);
fprintf(fileID,'xyplot 1 legendlabel ""\n');
fprintf(fileID,'xyplot 1 legend off\n');
fprintf(fileID,'xyplot 1 minorgrid on\n');
fprintf(fileID,'xyplot 1 title "EnergyIndentation_%s"\n',studyname);
% Saving finished plot
fprintf(fileID,'xyplot 1 savefile curve_file "%s\\Energyplot_%s" 1
all\n',loc,studyname);
% Printing finished plot to .jpg format
fprintf(fileID,'xyplot 1 select 1\n');
fprintf(fileID,'xyplot 1 show select\n');
fprintf(fileID,'xyplot 1 xmin -100 \n');
fprintf(fileID,'xyplot 1 xmax 3680\n');
fprintf(fileID,'xyplot 1 xoffset -100\n');
fprintf(fileID,'print jpg "%s\\%s.jpg" LANDSCAPE nocompress gamma 1.000 transparent
0x000000 enlisted "PlotWindow-1" \n',loc,studyname);
fprintf(fileID,'removefile "%s\\Energyplot_%s" 1 all\n',loc,studyname);
fclose(fileID);

end
```