MASTER'S THESIS

FINITE ELEMENT MODELLING AND PARAMETRIC STUDIES OF SEMI-CLOSED THIN-WALLED STEEL POLYGONAL COLUMNS

-Application on Steel Lattice Towers for Wind Turbines-

Bona Ryan
2017

Master of Science (120 credits)
Civil Engineering

Luleå University of Technology
Department of Civil, Environmental and Natural Resources Engineering

LULEÅ UNIVERSITY OF TECHNOLOGY

FINITE ELEMENT MODELLING AND PARAMETRIC STUDIES OF SEMI-CLOSED THIN-WALLED STEEL POLYGONAL COLUMNS

-Application on Steel Lattice Towers for Wind Turbines-

Bona Ryan

Supervisors: Efthymios Koltsakis
Panagiotis Manoleas

Luleå University of Technology
Department of Civil, Environmental and Natural Resources Engineering
Division of Structural and Construction Engineering - Steel Structures
SE - 971 87 LULEÅ
www.ltu.se/sbn

PREFACE

This thesis is a partial fulfillment of the requirements for the Master degree in Erasmus Mundus Master Programme Sustainable Construction under Natural Hazard and Catastrophic Events (SUSCOS_M). The work presented here was performed at the Department of Civil, Environmental and Natural Resources Engineering of Luleå University of Technology at the Division of Structural and Construction Engineering in the Steel Structure Research Group.

There are many people who had essential part in the creation of this thesis. The author would like to express his gratitude to the supervisor Prof. Efthymios Koltsakis, for his continuous supervision, guidance and support during this thesis work. Likewise, to Phd candidate Panagiotis Manoleas, a special word of appreciation, for his measureless help and invaluable feedback in the numerical modeling and development of this thesis, despite of the tight schedule of his own work. All their supports were deeply acknowledged.

I also would like to thank my SUSCOS colleagues here in Sweden for the time we spent together in the middle of extreme cold weather, for those related to the thesis and other stuff as well.

Special acknowledgements are addressed to programme coordinators Prof. Ing. František Wald from Czech Technical University in Prague, Prof. Luis Simoes da Silva from University of Coimbra, Prof. Dan Dubina, Ph.D. from Polytechnica University Timisoara, Prof. Dr. Jean-Pierre Jaspart from University of Liege, and Prof. Dr. R. Landolfo from University Federico II Naples, and all the people who involved in the Consortium of Erasmus Mundus SUSCOS Master Programme. This program, that connects six European technical universities together, has created a huge opportunity for me to meet and learn a lot from European most recognized steel experts. It is a big privilege to be a part of research focused on steel structure engineering.

Finally, I would like to express my sincere gratitude to my family, girlfriend and friends whose support have been always available and who helped me during this period of my life, even from distance, with kind words, encouragement, and infinite love. To my family I dedicate this thesis.

ABSTRACT

The trend of structural engineering in the recent years is toward the use of lighter and more economical structural elements. In steel construction, peculiarly, thin-walled structural elements are becoming more popular and have a growing importance. Improved techniques in a manufacturing of thin-walled elements have led to increased competitiveness of such products in the building applications. Some examples of such structural element can be found in everyday life in form of columns, studs, roofing trusses, and light-weight frames. However, the use of slender profiles and a complex cross sections shape lead to requirements to study instability phenomenon in a form of local, distortional, flexural, torsional and coupled instability. Such complex structural behavior is inevitably accompanied by demand to improve calculation methods and design provisions. In this thesis, an innovative solution of structural element composed of thin-walled plates is proposed for the application on lattice support structure of wind turbine.

Thin-walled cold-formed profiles are steel products usually made from cold rolled coils and folded in the second step. In this way, only open profiles can be produced. The predominant problem of the open cross-section is the excessive torsional effect caused by the non-coincidence between the shear center and mass center, and a poor torsional resistance. A better response is possible with closed cross sections, but such profiles could not be produced by the folding. The solution is to make semi-closed section by assembling them into polygonal profiles with mechanical fasteners, as presented in this thesis.

Objective of this work is to study the proposed structural sections in design situation and to investigate possible design models. The expected structural behavior of the column is a mixture between the open and closed cross-section. These cases will be investigated through numerical study. In this thesis presented a comprehensive parametric study on the ultimate strength of proposed cold-formed steel columns using the Finite Element package ABAQUS. FE models were first developed for columns by using automation that was made through MATLAB and Python script. The buckling and non-linear FE study was done for the investigation of local (L), distortional (D) and global (G) possible buckling failures and ultimate resistance, respectively. Modelling issues such as boundary conditions, meshing, initial imperfections, material models, and non-linear solution controls in FEA were also addressed.

The parametric study involved series of profiles of built-up polygonal cross-section types with varied thickness (t), number of corners (n), diameter (d), slenderness (*slend*), yield strength (f_y), number of points along corner radii (n_p), extension lip length (l_{ext}), gusset plate thickness (t_g), member length (l), and density of fasteners (*s/d ratio*), loaded in compression and bending moment. The bending moment occurs as the effect of forces acting on the connection. The purpose of this analysis is to study the critical load, cross-sectional behavior, influence of the amplitude of initial imperfections on the ultimate load, and influence of each parameter used in the analysis through Full Factorial Design.

Key Words

Lattice tower, Thin-walled element, Semi-closed polygonal section, Finite element analysis, Parametric study, Buckling, Python script, Matlab script, Eurocode 3, Factorial design

NOTATIONS

Latin capital letters

A_g	Gross cross sectional area	[mm^2]
A_{eff}	Effective area	[mm^2]
D	Diameter	[mm]
E	Modulus of elasticity	[N/mm^2]
I	Second moment of inertia	[mm^4]
I_{eff}	Effective second moment of inertia	[mm^4]
L	Length	[mm]
L_{cr}	Critical length	[mm]
$N_{b,Rk}$	Characteristic buckling resistance	[kN]
N_{cr}	Critical load	[kN]
N_d	Designed load	[kN]
N_{el}	Elastic load	[kN]
$M_{c,Rd}$	Characteristic resistance Moment	[kN]
M_d	Designed moment	[kN]
W_e	Elastic section moduli	[mm^3]
W_{eff}	Effective elastic section moduli	[mm^3]

Latin small letters

b	Width of a stiffened or unstiffened plate	[mm]
b_{eff}	Effective width	[mm]
b_p	Notional flat width	[mm]
f_y	Yield strength	[N/mm^2]
f_{ya}	Average yield strength	[N/mm^2]
f_{yb}	Basic yield strength	[N/mm^2]
f_u	Ultimate tensile strength	[N/mm^2]
i	Radius of gyration	[-]
r	Radius	[mm]
t	Thickness	[mm]

Greek small letters

$\bar{\lambda}$	Relative Slenderness	[-]
$\bar{\lambda}_p$	Plate slenderness	[-]
σ_{el}	Elastic compressive stress	[N/mm^2]
σ_{cr}	Critical buckling stress	[N/mm^2]
ε	Strain	[-]
k_σ	Buckling factor	

TABLE OF CONTENTS

1 INTRODUCTION

At current stage of wind converter development, reducing the amount of input cost is still a big challenge. Wind energy is positive in the way that the wind resource is renewable and it does not produce harmful impact like greenhouse gas from conventional power source. However, one of the shortcomings with wind power generation is that the amount of investment cost makes it not as much commercially competitive as other conventional power generation sources for the time being. Efforts are being carried out to optimize it. As for optimization of structural form, lattice system is proposed as support structure for wind turbine. Lattice structure possesses advantages in that it generally requires less material and the wind load impact is also reduced due to the reduced impacted area compared with monopole structure. In this part, lattice structure composed of steel semi-closed polygonal cross-section made by cold-formed thin-walled plates is proposed for more efficient structural solution, and compared to other common type of wind turbine towers. This novel built-up structural section is expected to deliver robust structure with economical feature.

1.1 Background

Wind power is considered one of the most promising alternative energy resources for production of electricity. One major advantage that wind power offers compared to conventional ways of producing energy such as fossil energy and nuclear energy is the low emission of carbon dioxide CO_2 during production of electrical energy. It is a clean renewable energy. In the last decades extensive research and huge resources have been focused on production of wind power around the world, especially in the European Union (EU) where annual wind power installations have increased steadily over the past 15 years from 3.2 GW in 2000 to 12.8 GW in 2015, an annual growth rate of over 9% [**1**].

A wind power station consists of a supporting tower structure with a turbine nacelle at the top. The cost of the tower is one of the most important aspects to address at the moment of design stage and installation. Based on experience, it covers approximately 20%of the total manufacturing cost for a wind turbine [**2**]. For turbines with higher rated power capacity, the percentage could be even increasing. Moreover, the height of the tower is an important factor in the efficiency of using wind power. Higher tower solutions have a significant role in reducing the unit costs of generated electricity [**3**], likewise latest development in wind power industry includes the effort to achieve larger blades radius and consequently higher tower. Building higher tower increases the output of wind power as the wind becomes more constant, less turbulence and the wind speed increases with the increase of the tower height. Hence, the optimization of supporting tower for more economical solution while keeping the structural performance is necessary.

Reduction of cost could be made through various methods: optimization of structural form can save material cost if sufficient structural strength is maintained or manufacturing cost can be lowered by means of mass production. The latter is in fact the reason why large scale wind farm development is becoming the interests of many. One possibility to reduce costs of tower manufacturing could be to produce them in smaller parts that are easier and less expensive to transport. For such a solution to be

feasible it is important that the resulting structure and connections has the required stability and load resistance. The problem of transportation of tower to inland sites and erection method significantly decreases if the tower is built by sections instead of large cross-section and in long segments. Moreover, the price of the tower can also be decreased by the use of thin elements with high strength values.

In the current design of steel towers for onshore wind turbines the most common type of tower is cylindrical tubular tower. Some challenging limitations regarding tower height and erection process attributed to this type of structures including transportation restriction for maximum shell diameter of 4 – 4.5m, fatigue endurance due to transversal and longitudinal welding, connection problems with thick flanges, expensive rolling process, and lifting technology that limit the height of current installed tower to be 80 – 100m.Lattice or truss support structure then proposed to deal with such problems, however the steel CHS section as common cross section for this type of structure for high onshore wind turbine has complication in welding connection and member thickness. All these issues will eventually affect the cost effectiveness of tower component in the overall building cost. Many researches have been looking for different solutions to these problems. Three projects which have been dealing with these questions are the HISTWIN [**4**], HISTWIN2 [**5**], and the on-going AEOLUS4FUTURE [**6**]. New challenging load conditions and new type of support structures for wind energy converter which foster new structural concepts and high performance material became the focus of those researches. As part of these projects, Heistermann [**7**] proved a new solution with friction connections to replace the traditional in flange connection, and then Garson [**8**] conducted comprehensive study of polygonal tubular towers made of folded plates. In 2008 the steel supplier "Rautaruukki Oy" [**3**] developed a method to build a lattice tower of 160 m using a new section type made from cold-formed steel as illustrated in Figure 11. The latter type of structural member is adopted in this thesis.

Besides the structural performance of member, one important issue for the stability of the presented tower is to analyze the connections required to maintain the structural performance. As the joint behavior affects distribution of both internal forces and moments as well as deformations of the structure, its investigation is very important. Therefore, the work presented here also addressed questions related to the structural characteristic of bolted connection on the lattice column made of polygonal thin-walled folded plates. Globally, this work is intended to promote competitiveness of semi-closed polygonal cross section for truss structure application.

1.2 Objectives and Expected Research Achievement

The main objective of this thesis is to study the structural behaviour of semi-closed polygonal column subject to various geometrical configuration and loading schemes. Another aspect with regard to the connections on this type of member is also considered. As of structural joints are characterized by means of their resistance, stiffness and rotation capacity, one parameter investigated here is estimation of stiffness of the connection under prescribed set of parameters.. The overall objectives are achieved by performing an extensive numerical parametric study to analyze and investigate the influence of different parameters on the response of the member.

The following key research questions are addressed:

1. Are the existing Eurocode EN1993 part 1-3 [**9**]analytical models for cold-formed steel sections suitable for predicting the critical axial force and bending moment on the proposed semi-closed polygonal section with bolted connection?

2. Is it possible to accurately predict the behavior of the proposed cross-sections as a built-up member by means of Finite Element Methods (FEM)? What are differences of such an approach compared to the analytical ones?

3. As for parametric studies, how far is the influence of each parameter on the structural response of the model?

4. Is it possible to investigate the influence of imperfection on the resistance of the proposed cross-sections?

1.3 Limitation

This thesis is endeavoring to solve the challenges for application of a new type of truss member as support structure of wind turbine and the purpose is to achieve a balance point between the safety, cost and the overall performance. Limitations exist because there is little existing experience and research to learn from for this thesis topic and many resources in the steel thin-walled industry are still not accessible to public. The whole thesis work is majorly based on conceptual studies and numerical analysis whereas there is a lack of verification from either physical modeling or industrial experience. The thesis is therefore trying to produce a work within its scope that could be useful or applied as reference when future relevant projects are to be planned.

The study performed in this thesis deals with the structural behavior of steel semi-closed polygonal sections with bolted connection in lattice structure, with focus on buckling behaviour, ultimate resistance and connection stiffness. In the numerical analyses, different types of structural configurations have been considered. Several parameters are applied on the parametric study to identify the influence on the response of the member. Thus, the analysis is limited to these specific prescribed parameters. The studied members in this thesis are series of segment of corner chord on lattice tower.

1.4 Methodology

To address the objectives and answers of the research questions in this thesis the following research methodology was adopted:

1. In the first step, a literature review has been carried out to identify the status of existing research work related to cold-formed polygonal section. The existing rules according to Eurocode EN1993 are briefly presented. Results from other researches are also presented. A particular focus was on built-up section made of thin-walled folded plates with bolted connection.

2. Thereafter, automation was carried out to build 3D finite element models of the sub-assemblies for parametric studies. MATLAB code was used to generate profiles of model and continued with automation in Python. Focus is given to numerically study and investigate the influence of different parameters on the response of the section.

3. After evaluating the results, focus is shifted to the stiffness prediction of connection on the studied columns. In addition, influence of imperfection is also investigated.

4. An evaluation of the results of the models performed by FEM and recommendation for the applicability of numerical calculation for this type of cross-section with regard to the standards of Eurocodes for steel design: EN 1993 part 1-3 [**9**] and EN1993 part 1-5 [**10**] was done.

1.5 Organization of the Thesis

The thesis consists of seven chapters excluding annexes. A brief summary of the content of each chapter is explained as follows.

Chapter 1 gives a background and a brief introduction of the research subject. Therein, also present the objectives and expected research achievements, limitations, and research methodology.

Chapter 2 gives a state of the art review. It starts with an overview of the rules for the considered type of thin-plates as presented in Eurocode EN1993. In addition research results from other references are also presented.

Chapter 3 presents the procedure of parametric studies and numerical analysis. Detail description of FE models developed for the proposed semi-closed polygonal cross-section is described here.

Chapter 4 describes the evaluation of result from FE-calculations and parametric studies for the proposed polygonal cross-section.

Chapter 5 gives procedure for the numerical modeling and calculation of stiffness of the bolt connection along the member

Chapter 6 gives the investigation of moment-rotation behaviour of the gusset connection on the member

Chapter 7 sums up the main conclusions achieved and provide some suggestions for future work related to the research presented in this thesis.

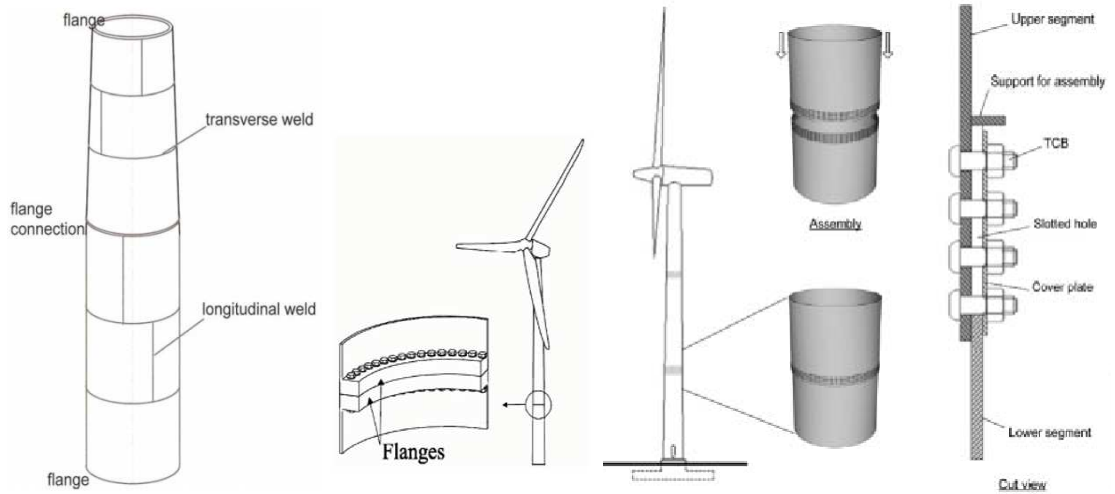
1.6 Type of Towers for Wind Energy Converter

1.6.1 Welded steel shell tower

The welded steel shell tower today dominates the wind turbine market. It consists of cylinders made of steel plate bent to a circular shape and welded longitudinally (Figure 1). Transversal welds connect several such cylinders to form a tower section. Each section ends with a steel flange in each end. The sections are bolted to each other. The bottom flange is connected to the foundation and the top one to the nacelle.

A tower is primarily dimensioned against tension and buckling in the extreme load cases. Ideally the margin should be the same for both criteria, since increasing the diameter, with a corresponding reduction of plate thickness, increases the tension strength but reduces the buckling margin. Finally the tower has to be checked against fatigue. According to Eurocode connecting welds (transversal and longitudinal) and dimension changes (flanges) affects the strength in a negative way. Thus it is the welds and the geometry that primarily determine the fatigue strength rather than the quality of the steel. Therefore wind turbine towers mostly use ordinary qualities of steel. In this report use of S355J2G3 (earlier known as SS2134, tensile yield limit 355 MPa) is assumed for both the welded and friction joint towers.

In the dimensioning load case, the tower is affected by the thrust from the rotor. This thrust will create a bending moment, which increases with the distance from the turbine shaft, i.e. inversely proportional to the height above the ground. To cope with this increasing bending moment it is favourable to make the tower conical in shape, to the limit of buckling. However, land transportation even with a special permit is not possible for diameters exceeding 4.5 m in Sweden. Other countries and certain roads may create even more severe restrictions, e.g. 3.5 m. To a certain degree these restrictions may be counteracted by an increase of plate thickness, however, the tower will then become less economical [**11**].



a) Shell tower in two sections **b) Ring flange connection** **c) Friction connection**

Figure 1.a) Modularized tubular sections with two different connection types: b) ring flange and c) friction connection proposed in HISTWIN project

The specific investment cost for the different alternatives is summarized in Figure 2. The intended higher hub height alternative was not possible to attain with the 4.5 m base diameter limitation. For the 5 MW turbine the limit was 100 m. For all towers the maximum plate thickness is 75 mm. According to one source, some manufacturers experience difficulties above 50 mm.

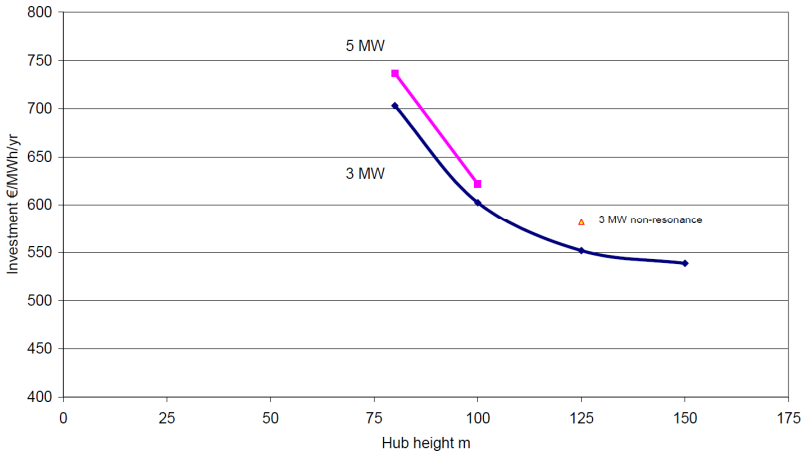


Figure 2. Summary of specific investment cost for 3 and 5 MW wind turbines furnished with welded steel shell towers, maximum diameter 4.5 m [11]

Besides making the tower expensive, a small tower diameter also means difficulties with transferring the loads into the foundation and also with the distribution of the loads in the foundation [**12**].

1.6.2 Pretensioned Concrete Tower

In a concrete tower the concrete proper only withstands pressure. The ability to absorb tension is provided primarily by pretensioned tendons, located in ducts in the concrete or internal/external of the concrete walls. Putting them internal or external enables easy inspection. There are also traditional untensioned reinforcement bars cast into the concrete shell, necessary to provide the compressive strength.

A concrete tower is clearly dimensioned by the extreme load case, since it has large margins towards fatigue. It is assumed that the concrete is pretensioned by the tendons to 20 MPa. In the extreme load case the pressure side is offloaded to close to zero whereas the tension on the other side is doubled. By increasing the thickness of the concrete cover it may be possible to increase the lifetime to e.g. 50 years. One concrete tower may then serve for two generations of machineries, with obvious economical savings (Figure 3).

Compared to steel towers, concrete towers are much heavier and takes longer time to erect. On the other hand, the concrete or the concrete elements, if made small enough, are not subject to transportation restrictions, as for the case with welded steel towers with large base diameters.

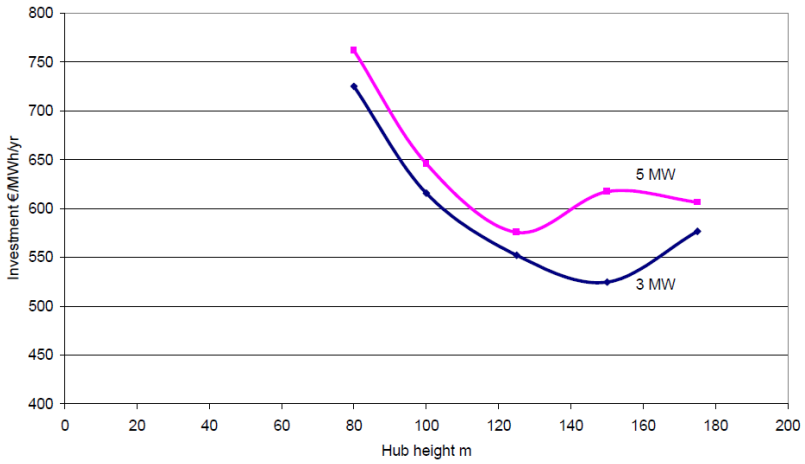


Figure 3. Summary of specific investment cost for 3 and 5 MW wind turbines furnished with slip formed concrete towers [11]

Regardless if the tower is slip formed or assembled from precast elements, it is advantageous to install the post-stressing tendons from below, thus not needing to lift the heavy rolls of tendons to the tower top. Then it is however necessary to furnish the foundation with a cellar [**12**].

1.6.3 Concrete/Steel Hybrid Tower

The idea behind building a hybrid concrete/steel tower is to use concrete in the wide lower part and steel in the upper part, where a conventional welded steel shell tower section may be designed without any risk of conflict with the transportation limitations. In reality it also makes it easier to design the concrete part and to get the eigen-frequencies right [**11**].

Figure 4. Concrete-steel hybrid tower

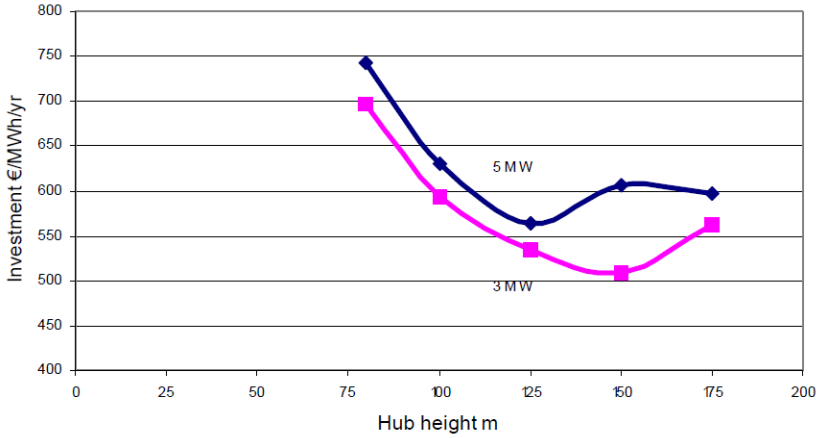


Figure 5. Summary of specific investment cost for 3 and 5 MW wind turbines furnished with hybrid concrete/steel towers[11]

1.6.4 Lattice Tower

The simplest construction method of building higher and stiffer towers is a three-dimensional truss, so called lattice or truss towers [**13**]. Lattice towers have been used in large numbers for smaller wind turbines. For larger turbines they have mainly been a choice when a stiff (under-critical) tower was needed. They are typically manufactured by means of welded or bolted tubular steel profiles or L-section steel profiles. The lattice towers are typically three- or four- legged and consist of corner chords interconnected with bracings in a triangulated structure.

It is clear that they often are considerably lighter than towers based on other technologies. The physical background to this phenomenon is the large widths of the lower sections. The need for material to take strain or pressure is inversely proportional to the width. With a tubular section a thin-walled construction will finally meet with buckling, which restrains the maximum diameter. A lattice design does not buckle like a shell. The risk of buckling of the individual members is controlled by inserting numerous struts that give the lattice tower its characteristic look.

The basic advantage of lattice towers is cost, since a lattice tower requires only half as much material as a freely standing tubular tower with a similar stiffness as already described above (Figure 7). As for optimization of structural form, the 20% or even higher cost of the tower could be possibly reduced if for instance lattice structure is applied as support structure for wind turbine. One another stated advantage of lattice towers is that they should have less aerodynamic drag and hence create less tower shadow and noise. On the other hand, transportation of lattice structure is also much more convenient when road transportation capacity of 4-6m is limited if a larger dimension of monopole is required for a larger scale wind turbine

Figure 6. Steel lattice tower

The visual qualities are controversial, especially due to the resemblance to towers for high-voltage power lines, generally claimed to be ugly. An open design, like a lattice tower, is more prone to icing than a tubular tower. The possible impact on the dynamic properties may be the most severe consequence, which may endanger the wind turbine in an extreme case. It may also be a problem for maintenance personnel, even if their elevator runs on heated rails. Another danger for such tower is the increased risk of falling ice. The last resort for evacuating a wind turbine nacelle is normally by a rope to the ground.

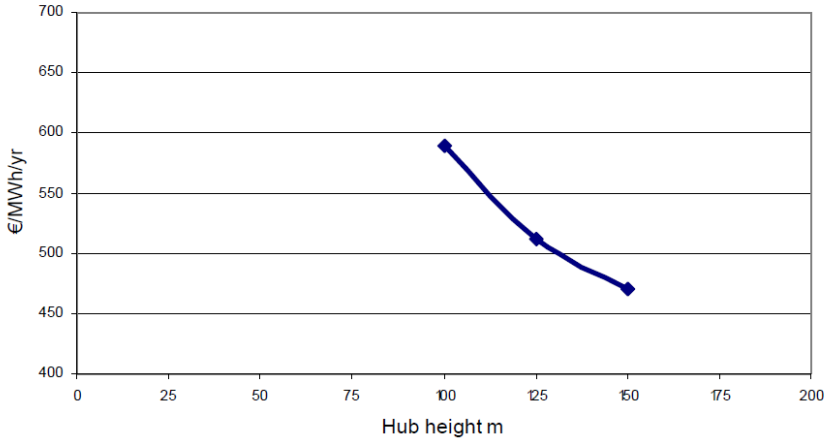


Figure 7. Specific investment cost for 3 MW wind turbine furnished with lattice towers

1.6.5 Hybrid Lattice-Tubular Towers

The most recent type of wind tower is hybrid lattice-tubular tower. This type of towers is intended to achieve greater hub heights. It consists of the three components, the lower lattice part, a transition piece, and tubular part. The lattice tower is set on the foundation and is assembled directly at site. The transition piece connects the lattice tower to the tubular tower, ensures the connection and transmission of forces between the two main parts. The tubular tower consists of sections joint by means of bolted flanges as happen in most tubular towers.

The tower can be climbed from the inside and is equipped with working platforms, ladder with fall protection system and a service lift. The bottom cabinet sections are arranged in the transition piece. The cabinet sections are connected to the generator and the top cabinet in the nacelle via power and control cables. At the tower top the cables are routed through a cable loop. It allows the nacelle to turn several times in each direction without damaging the cables. The power cables between WTG and grid are routed through conduits. The tower is protected against corrosion with special coating.

A tower of this type was installed at the wind farm in Gujarat, India (Figure 8). It is expected that this new type of towers produces about 10 to 12% more energy, because gains against the normal towers more than 40 meters in total height, with a combined height of 120 meters compared to the 80 meters of most tubular towers, therefore it is ideal for low wind areas, due to its superior performance, with a potential to be installed in all parts of the world, without having to look for places where the wind speed is high [**12**].

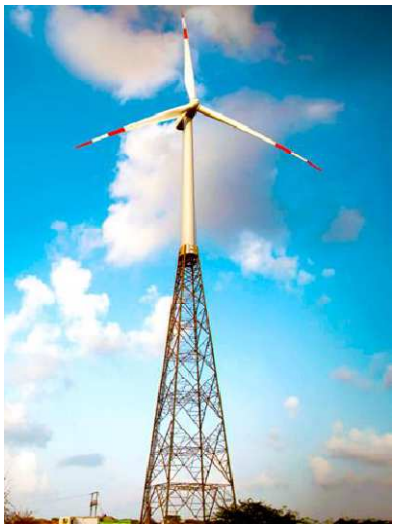


Figure 8. Steel hybrid lattice-tubular tower

1.6.6 Summary of Type of Support Structure

In the early experimental stage of wind energy and especially when the size of wind turbine was still moderate, the emphasis was not placed on cost reduction of the tower, which is why the tubular tower was very widely and popularly used. However, as commercialization of wind energy is urged and the size of wind turbine grows, after cost reduction measurements on mechanical components like the gearbox and generator are achieved, cost minimization associated with the turbine support structure is again attracting interests and this is why these years the lattice structure is receiving more and more attentions, especially on large scale wind turbines.

A comparison in terms of the investment cost of a commissioned wind turbine divided by the yearly production, called the specific cost for the presented types of tower is shown in Figure 9 below.

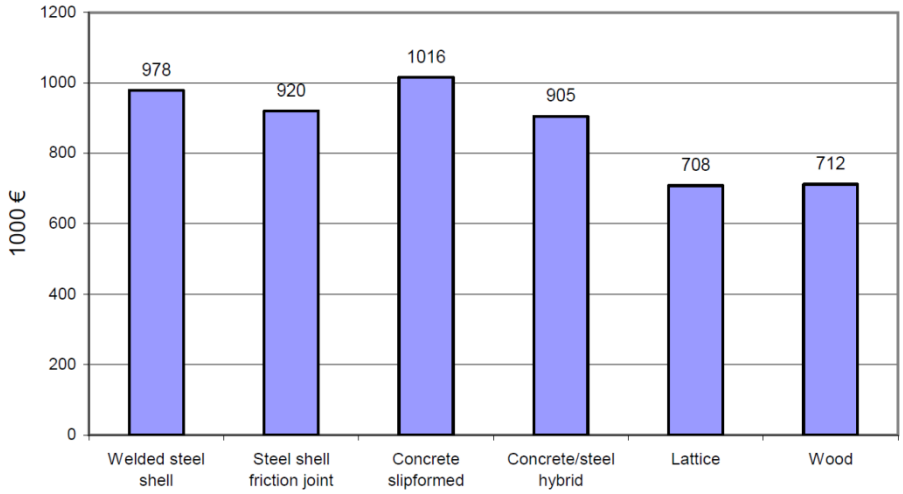


Figure 9. Tower cost for the alternative designs. Power 3 MW, hub height 125 m[11].

1.7 Semi-closed polygonal section trusses

The idea of using built-up polygonal profile member made of folded plates for the truss element is an innovative and recent solution, therefore the research and previous studies about this matter are not much available. The objective is to maximize the efficiency of the cross-section by its geometry, while minimizing the quantity of steel used. This can lead to great economical and environmental benefits. There is no extended research and literature behind this type of built-up hollow sections so far and therefore there are many uncertainties on how these elements would behave under loads.

The main advantage of semi-closed polygonal profiles made from cold-formed steel is that they facilitate simpler connections with minimum welding. Figure 10 shows possible polygonal profiles for compression chords and diagonals in a lattice. The gusset plates required for the connections are inserted into the polygonal profile and secured with pretension bolts.

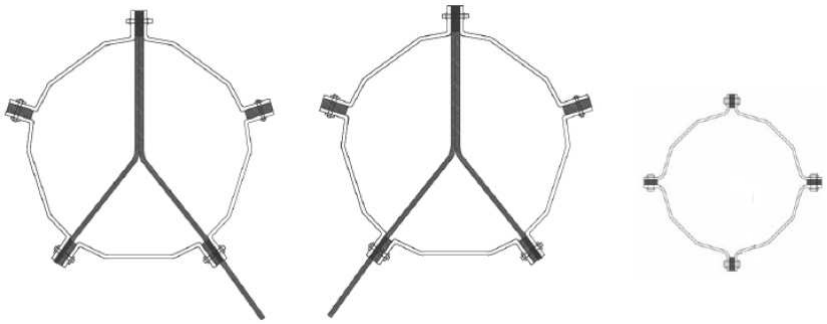
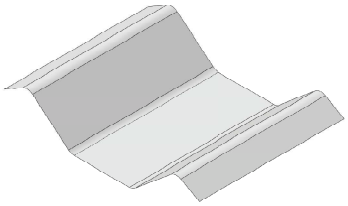


Figure 10. Illustration of connection of semi-closed polygonal cross sections chords to diagonals

A detailed research study about this type of section was performed in [**8**], which investigated the resistance of the polygonal cross-sections for tubular tower application. The focus of that research is the use of thin walls on bolted elements in wind tower applications and the assessment of the design methods according to Eurocodes in comparison with FEM analysis. The results of the study show that

the Eurocode EN1993 part 1-3 and part 1-6 are in a good agreement when compared to the laboratory tests and FEM analysis performed, whenever the axial resistance was done on the folded plates. A smaller difference between numerical and analytical results was obtained when calculating the critical load with EN1993 part 1-5, rather than with part 1-6. Therefore, in this thesis EN1993 part 1-5 is also used in order to determine the critical load of plates. It is also shown that the strength of the folded plate, even with less material used in the cross-section has a higher efficiency than the plates with circular cross-sections [**8**].

Furthermore, as part of HISTWIN project, lattice tower using a new section made from cold-formed steel as illustrated in Figure 11 was developed by Ruukki. A preliminary study by experimental test and FE analysis of the behavior of this new polygonal cross-section consist of folded plates was conducted at Luleå University of Technology [**8**].



(a)

(b)

(c)

Figure 11. Lattice (truss) tower using semi-closed polygonal cross section, a) Steel plate, b) tower section and c) Lattice tower designed by "Rautaruukki Oy"

This thesis is an extended research for built-up polygonal cross-section for application in lattice / truss structures with bolted connection. Several numerical models are developed for parametric study to understand the structural behavior and evaluate the efficiency of the proposed solution in comparison with the existing Eurocode for cold-formed steel.

2 THIN-WALLED STEEL PROFILES AND THEIR RESISTANCE ACCORDING TO EUROCODES

Cold-formed steel sections demonstrates extensive applicability, even it is a relatively new system, due to some great advantages, such as high strength-to-weight ratio, reduced labor costs and fast erection due to the light weight of cold-formed members. In the form of individual plates, they have an inherent weakness in their small torsional stiffness, which is unfavorable for compressive members. The solution is to make closed section by assembling them with fasteners. Cold-formed steel design is dominated by two specific problems, i.e. (1) stability behaviour, which is dominant for design criteria of thin sections, and (2) connecting technology, which is specific and influences significantly the structural performance and design detailing. The work presented in this chapter deals with resistance of the proposed section in relation with stability and connection, based on analytical approach in the Eurocodes.

2.1 Introduction

In steel construction cold-formed structural members are becoming more popular and have a growing importance. One of the reasons is the versatile nature which allows for the forming of almost any section geometry which can be produced at low-cost by cold forming and rolling from thin steel sheets. Cold–formed steel sections are usually thinner than hot–rolled sections and can be subject to different modes of failure and deformation and therefore extensive testing is required to provide a guideline for the design of cold-formed thin-walled structural members [**14**].

The main mechanical properties (yield point, tensile strength and ductility) of cold-formed steel sections, particularly at the corners, are considerably different from those of the flat steel sheet, plate, strip or bar before forming. This is because the cold–forming operation increases the yield point and tensile strength, and at the same time decreases the ductility. Design codes have been generated for cold-formed steel structures subjected to various loading scenarios which can cause buckling, bending and web crippling or a combination.

Cold formed sections can be optimized for different purposes and they are fairly inexpensive to produce in small series by brake forming. The first objective of the optimization is to avoid the detrimental effects of local buckling of compressed parts. This is done by folds and by forming intermediate stiffeners in wide flat parts. Most of the cold-formed members have open cross-sections with a very small torsional stiffness. This means that the resistance to global buckling frequently is governed by torsional or torsional-flexural buckling with a relatively low resistance compared to flexural buckling. One way of improving the resistance is to make the cross-section closed by using mechanical fasteners for the connection, which will be investigated in this thesis. It is here called a semi-closed cross-section because it is not continuously and rigidly connected.

Using the newest developments in material technology and in cold forming technique the processing of steel sheets with yield strength ranging from 250 MPa to 700 MPa and larger sheet thicknesses becomes possible. Due to that, trusses with cold formed sections show a marked reduction in weight in comparison to hot rolled sections. However, this advantage can only be used for structures, where serviceability is not decisive, which is the case for trusses with their large flexural rigidity. For conventional truss structures made from hot rolled sections, design details were optimised during decades of application, while for an economical design of trusses made of cold formed sections, new cross-section types as well as truss joint details have to be developed. In [**15**] a pentagon shaped cross-section is proposed and investigated through a calculation method based on the Generalized Beam Theory (GBT), which was compared to numerical calculations and experimental data. The main concepts and steps that need to be followed when developing the numerical implementation of a GBT formulation aimed to perform first-order elastic–plastic analyses of thin-walled members have been presented there. All the GBT results were compared to ABAQUS shell finite element value, very good agreement between the two being obtained. However, it is shown that FEM analysis provides better and more precise results than the GBT procedure and therefore, in this work an FEM approach is adopted.

The objective of this chapter is to assess the established design procedure of an axially compressed cold-formed steel member and stiffness of the connection. The investigated steel member is a segment of a semi-closed corner chord as illustrated in Figure 11. The element was intended to be used as part of a lattice tower for wind turbines. The resistance to axial compression of the folded plate was analysed according to the rules of EN-1993 rules part 1-3 [**9**] for the folded cold-formed members and then evaluated with the help of FEA.

2.2 Manufacturing of Cold-Formed Steel Section

Cold-formed members are steel products made from coated or uncoated hot-rolled or cold-rolled flat strips or coils. Within the permitted range of tolerances, they have constant or variable cross section. Individual structural members (bar members) obtained from so called “long products” including single open sections, open built-up sections, and closed built-up sections.

Cold-formed members are normally manufactured by one of two processes, namely:

- Roll forming;
- Folding and press braking.

Roll forming consists of feeding a continuous steel strip through a series of opposing rolls to progressively deform the steel plastically to form the desired shape. Each pair of rolls produces a fixed amount of deformation in a sequence of type shown in Figure 12a. Each pair of opposing rolls is called a stage as shown in Figure 12. In general, the more complex the cross sectional shape, the greater the number of stages required.

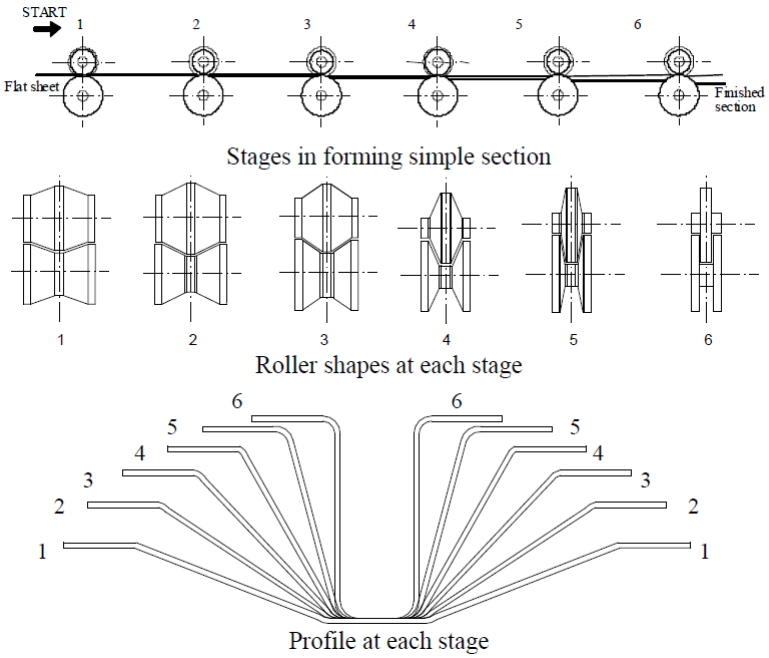


Figure 12.Stages in roll forming a simple section(Rhodes, 1991)

Folding is the simplest process, in which specimens of short lengths, and of simple geometry are produced from a sheet of material by folding a series of bends, as shown in Figure 13. This process has very limited application.

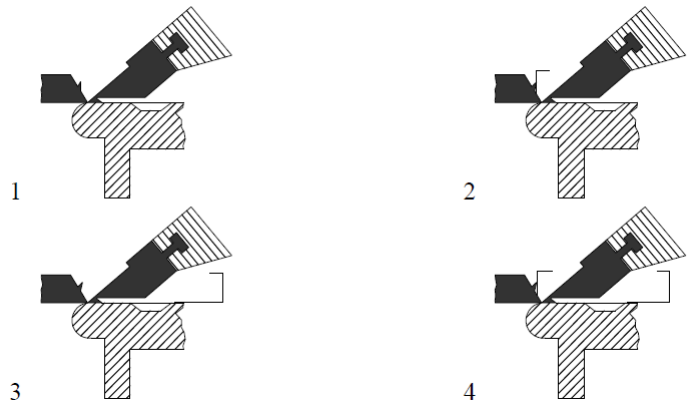


Figure 13. Folding process

Press braking is more widely used, and a greater variety of cross sectional forms can be produced by this process. Here a section is formed from a length of strip by pressing the strip between shaped dies to form the profile shape, as shown in Figure 14. Usually each bend is formed separately. This process also has limitations on the profiled geometry which can be formed and, often more importantly, on the lengths of sections which can be produced. Press braking is normally restricted to sections of length less than 5 m although press brakes capable of producing 8 m long members are in use in industry.

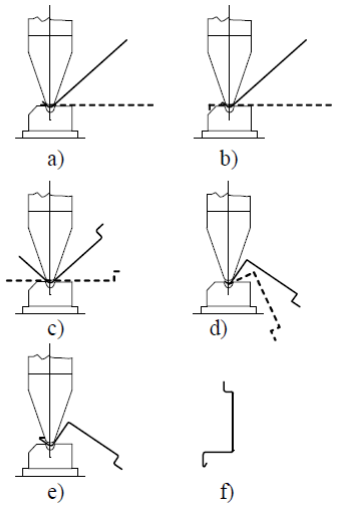


Figure 14. Press braking process

2.2.1 Peculiar Characteristic of Cold-Formed Sections

Compared to hot-rolled steel sections, the manufacturing technology of cold-formed steel sections causes some peculiar characteristics. First of all, cold-forming leads to a modification of the stress-strain curve of the steel. With respect to the virgin material, cold-rolling provides an increase of the yield strength and, sometimes, of the ultimate strength that is important in the corners and still appreciable in the flanges, while press braking leave these characteristics nearly unchanged in the flanges.

Table 1. Influence of manufacturing process on the strengths of hot and cold-formed sections

Forming method		Hot rolling	Cold forming	
			Cold rolling	Press braking
Yield strength	Corner	--	high	high
	Flange	--	moderate	--
Ultimate strength	Corner	--	high	high
	Flange	--	moderate	--

The increase of the yield strength is due to strain hardening and depends on the type of steel used for cold rolling. On the other hand, the increase of the ultimate strength is related to strain aging, which is accompanied by a decrease of the ductility and depends on the metallurgical properties of the material. Eurocodes provide formulas to evaluate the increase of yield strength of cold-formed steel sections, compared to that of the basic material.

Hot-rolled profiles are affected by residual stresses, which result from air cooling after hot-rolling. These stresses are mostly of membrane type, they depend on the shape of sections and have a significant influence on the buckling strength. In the case of cold-formed sections the residual stresses are mainly of flexural type (Figure 15), influenced by cold rolling procedure, and their influence on the buckling strength is less important than membrane residual stresses as shown in Table 2.

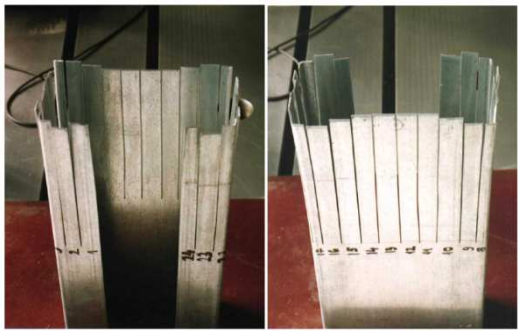


Figure 15.flexural residual stresses in a lipped channel cold-formed steel section

Table 2. Residual stress in steel sections

Forming method	Hot rolling	Cold forming	
		Cold rolling	Press braking
Membrane residual stresses (σ_{rm})	high	low	low
Flexural residual stresses (σ_{rf})	low	high	low

The fact that the mechanical properties of cold-formed sections, i.e. yield strength and residual stresses – are different to those of hot rolled ones, different buckling curves need to be justified. However, for the simplicity of the design process, the same buckling curves as for hot formed sections are still used in Eurocodes.

2.3 Buckling Behaviour

2.3.1 Elastic buckling of column

Buckling is a geometric non-linear problem. Buckling of a member occurs when the axial force in the member is so high, that the member cannot resist this axial force in combination with lateral deflection and thus no stable equilibrium can be found. The value for which this happens is called a bifurcation point because an intersection between two equilibrium paths is present. For the ideal column (a perfectly straight column is subjected to a centrically applied load) when the load is less than the critical load, the column will remain straight in state of stable equilibrium. When the load is continuously increased equilibrium is not only possible for a straight column in deformed shape, however this is state of unstable equilibrium meaning that small disturbances leads to large displacements. In Figure 16 is shown in a force-deflection diagram. The vertical axis represents the axial compressive force in the column (hinged at both ends) and the horizontal axis shows the lateral deflection in the middle of the column.

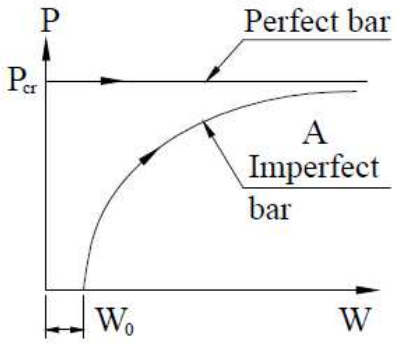


Figure 16. Euler buckling load

In a Linear Buckling Analysis, it is assumed that the column is centrically loaded, the material is homogeneous, isotropic and perfectly elastic and that the deflections are small. A maximum compressive stress is not taken into account. This way, a buckling load can be found by finding an equilibrium between the compressive force multiplied with an eccentricity (which is only present if buckling occurs) and the flexural stiffness of the column where the axial compressive force is called the disturbing force and the force caused by the flexural stiffness of the column is called the internal resisting moment. The disturbing force and the internal resisting moment must be in equilibrium for the column to be stable. Using a differential equation to describe its behavior, the first order or linear buckling load can be found by equation below, which is found by solving the differential equation.

$$N_{cr} = \frac{\pi^2.E.I}{(K.L)^2}$$

where:
N_{cr} is the critical load or buckling load of the column
EI is the flexural stiffness of the column
L is the system length
K is effective length factor

And critical stress is defined by:

$$\sigma_{cr} = \frac{\pi^2 E}{\lambda^2} = \frac{N_{cr}}{A}$$

where:
σ_{cr} is the critical stress, the stress whereby the column buckles
λ is the slenderness of the column
A is the area of the section

$$\lambda = \frac{L_{cr}}{i} = \sqrt{\frac{\pi^2 E}{\sigma_{cr}}}$$

where:
i is the radius of gyration, determined by

$$i = \sqrt{\frac{I}{A}}$$

With the above equations a graph can be plotted with the critical stress on the vertical axis and the slenderness on the horizontal axis (Figure 17).

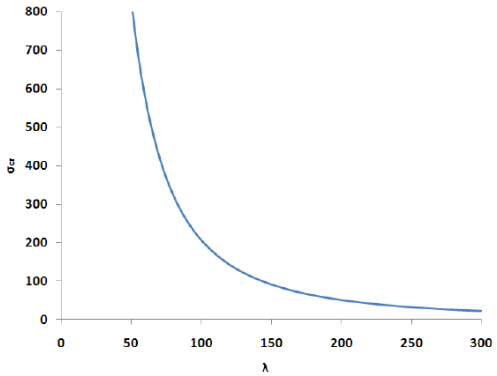


Figure 17. Euler buckling curve

It can be seen that the critical stress increases when the slenderness decreases. The stress even exceeds the yielding stress of steel. Of course, this is impossible, but as outlined before, it is assumed the material is perfectly elastic and a maximum stress is not taken into account.

The failure of a column under compressive stresses could be either local or global buckling. In the case of global buckling the deformation can be in bending, torsion or in combination depending on the type of cross-section. Local buckling occurs in the form of one or several small buckles. An important characteristic of local buckling is the post-buckling range, as the load can be increased beyond the load at which an initially flat cross-section buckles. The process of local buckling and global buckling is different depending on the type of structural element (Figure 18).

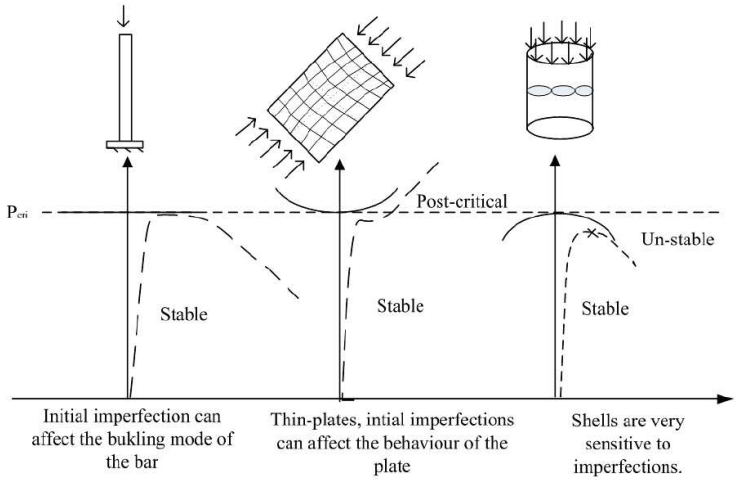


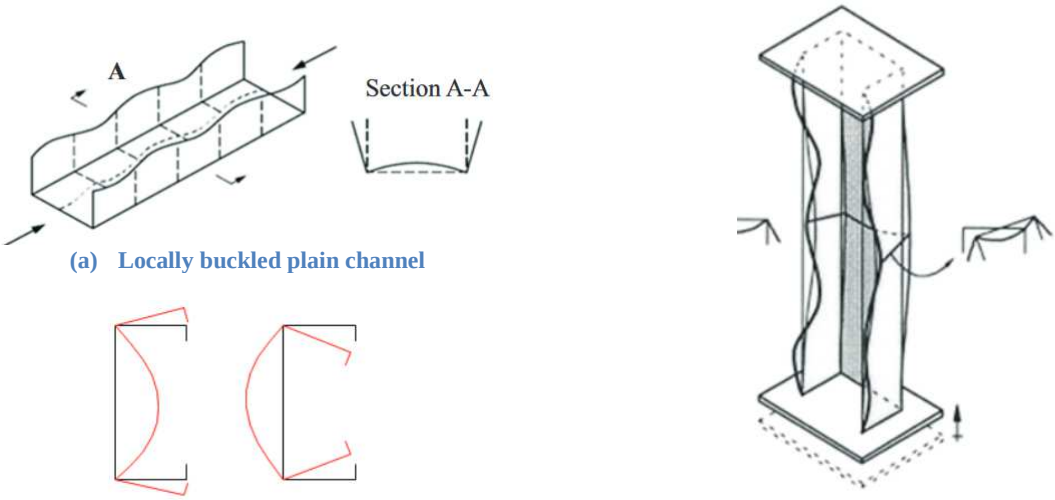
Figure 18. Stability of a member under axial compression

It can be seen in Figure 18 that compared to other kinds of structures, thin plates are characterised by a stable post-critical behaviour. Standards such as EN 1993 part 1-5 consider the susceptibility of structural element to buckling by the classification of the cross-sections. Where Classes 1 and 2 is used for elements where buckling less possible to occur. Class 3 and 4 is used when the slenderness is higher and buckling occurs before the yield stress is reached. Cross-sections formed from thin plates usually fall in class 4 and are very sensitive to imperfections under axial compressive loads.

2.3.2 Buckling behaviour of cold-formed structural members

The use of thinner sections and high strength steels leads to design problems for structural engineers which may not normally be encountered in routine structural steel design. Structural instability of sections is most likely to occur as a result of the thickness of the sections, leading to reduced buckling loads (and stresses), and the use of higher strength steel typically makes the buckling stress and yield stress of the thin-walled sections approximately equal.

Unlike heavy hot-rolled steel sections, cold-formed thin-walled sections tend to buckle locally at stress levels lower than the yield strength of the material, characterised by the relatively short wavelength buckling of individual plate element. However, these members do not fail at these stress levels and continue to carry further loads. This phenomenon is called post-buckling behaviour, as shown in Figure 18. In thin-walled cold-formed steel structures, elastic bucking and load deformation response are closely related. There are different buckling modes as illustrated in Figure 19: Local [**16**], Distortional [**16**], Euler (Flexural), Torsional–Flexural of column or Lateral–Torsional of beam [**17**].



(a) Locally buckled plain channel

(b) Distortional buckling mode

(c) Flexural-torsional buckling mode

Figure 19. Different buckling modes

It is well known that cold-formed thin-walled members under compression load exhibit local and global post buckling behavior [**18**]. Overall column strength is greatly dependent on the interaction effect of local and overall column buckling. The curves shown in Figure 20 have been obtained using elastic Finite Strip (FS) software, analysing and describing the change of buckling strength versus buckle half-wavelength.

It is recognised that cold formed steel compression members may locally buckle before the applied load reaches the overall collapse load of the column. The influence of local buckling on column strength depends on the following factors: the slenderness ratio of the column, the shape of the cross section, influence of cold work, the type of steel used and its mechanical properties, the type of governing overall column buckling, effect of imperfection, effect of welding, interaction between plane components, effect of perforations if any, effect of residual stress etc. For short columns, local buckling occurs first leading to large deflections. Local buckling involves deformation of the component plate elements of the section, with the plate junctions remaining straight and occurs as

plate buckling of individual slender elements in a cross-section as shown in Figure 19(a),identified by first minimum buckling stress at Point A (Fig. 20).

Distortional buckling is characterised by rotation of the flange at the flange/web junction in members with edge stiffened elements and occurs only for open cross-sections where the compressed flanges buckle inward or outward along the length of a member as shown in Figure 19(b). This is the form of a ”semi–local” buckling mode which can generally arise at somewhat longer wavelengths, generally intermediate between that of local buckling and global buckling. The distortional buckling mode has been found to govern the strength of sections with intermediately stiffened elements [**16**], [**17**]. The elastically determined post–buckling behaviour is generally stable, but in–plane deformations can encourage substantial membrane stresses and rapidly produce yield and failure in the stiffeners [**19**], [**20**].

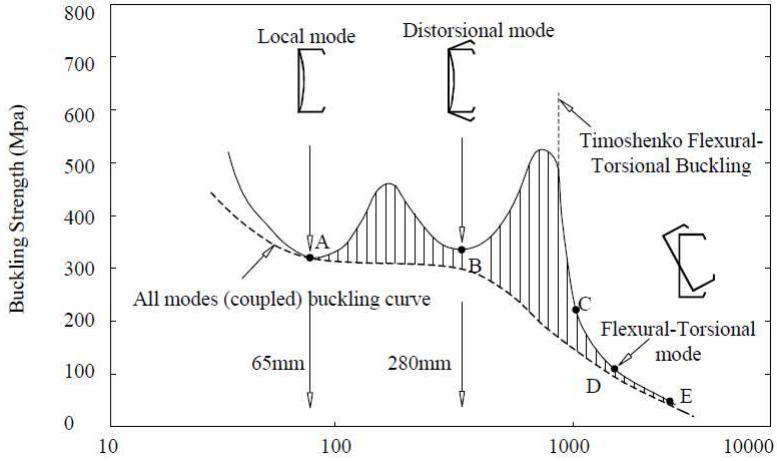


Figure 20.Buckling strength versus half-wavelength for a lipped channel in compression (Hancock, 2001)

For long columns under compression load, Euler buckling is more likely to occur before any other instability failure and occurs by bending about the web[**21**]. When a slender member is subjected to an axial force, failure in the member takes place due to torsion or bending rather than direct compression of the material. Further, if the compression member is not supported in the lateral direction the member will fail due to lateral buckling of the compression flange [**22**].This phenomenon occurs when the flexural load increases to a critical limit. In wide–flange sections, if the compression flange buckles laterally, the cross section will also twist in torsion, resulting in a failure mode which represents lateral–torsional buckling. The lateral torsional buckling strength of a section could be increased by using a bracing system. In general, bracing systems could be divided into two categories namely, lateral bracing and torsional bracing. Cold formed steel structures are made of thin steel sheets and have often mono symmetric or un–symmetric cross sections. Hence their lateral–torsional buckling behaviour is more complicated than that of doubly symmetric hot–rolled beams. Past research on lateral–torsional buckling of steel beams has mainly concentrated on hot–rolled steel beams [**23**].

The dashed line in Figure 20, qualitatively shows the pattern of all modes or coupled mode. The effect of interaction between sectional and global buckling modes results in increasing sensitivity to imperfections, leading to the erosion of the theoretical buckling strength.

In general, the buckling mode is influenced by cross–section geometry, end conditions, loading and material. Also, introducing openings and imperfections to structures has a significant effect on the critical buckling load and the buckling mode [**24**], [**25**].

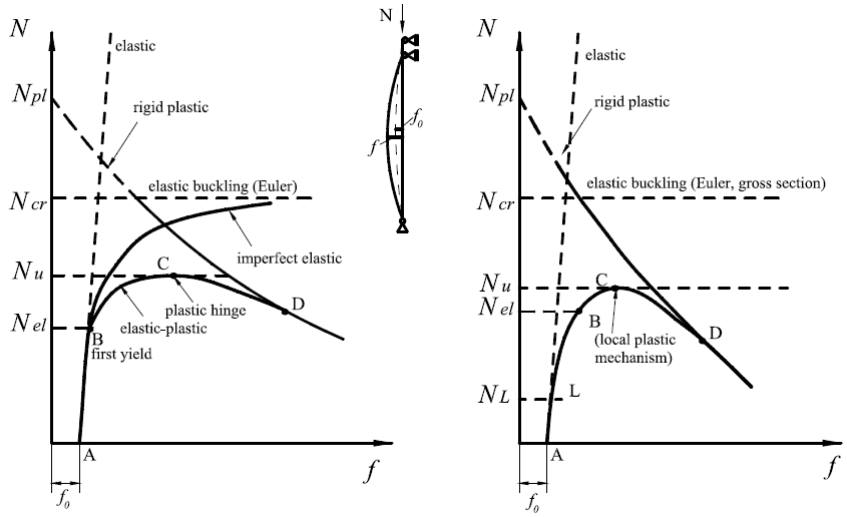


Figure 21. Behaviour of (a) slender thick-walled and (b) thin-walled compression column

Figure 21 shows the difference in behaviour of a thick-walled slender column in compression and a thin-walled column. It shows the behaviour of a thick-walled bar that it begins to depart from the elastic curve at point B when the first fibre reaches the yield stress, N_{el}, and it reaches its maximum (ultimate) load capacity, N_u, at point C; after which the load drops gradually and the curve approaches the theoretical rigid-plastic curve asymptotically. The elastic theory is able to define the deflections and stresses up to the point of first yield and the load at which first yield occurs. The position of the rigid-plastic curve determines the absolute limit of the load carrying capacity, above which the structure cannot support the load and remain in a state of equilibrium.

In case of a thin-walled bar, sectional buckling, e.g. local or distortional buckling, may occur prior to the initiation of plastification. Sectional buckling is characterised by the stable post-critical path and the bar does not fail as a result of this, but significantly lose stiffness. Yielding starts at the corners of cross section prior to failure of the bar, when sectional buckling mode changes into a local plastic mechanism quasi-simultaneously with the occurrence of global buckling.

2.3.3 Coupled Instabilities

In many practical cases, as it is for thin walled members, the loss of stability is often complicated by the occurrence of two or more eigenmodes at coincident or nearly coincident critical loads. This leads to an increase in sensitivity to geometric imperfection with reduction (erosion) of the expected buckling strength.

The natural effect of geometrical imperfections is the erosion of the cusp formed by the interaction of the pre-critical and post-critical paths. In the case of coupled instabilities (termed herein used to express the simultaneous occurrence of different buckling modes) further reduction is produced.

- Nature of the Phenomenon

The occurrence of simultaneous buckling arises in practical situations when the attempt is done to optimize the structure by choosing a specific geometry that allows the buckling modes to have the same critical load: this is the so-called “simultaneous buckling design principle”.

The matter is to analyse which are the effects of one buckling mode on the other; the interaction between the two modes generates a change in the buckling shape causing severe imperfection sensitivity and modifying the apparent optimization assumed with this approach.

The phenomenon can be illustrated by a simple model as shown in Figure 22.

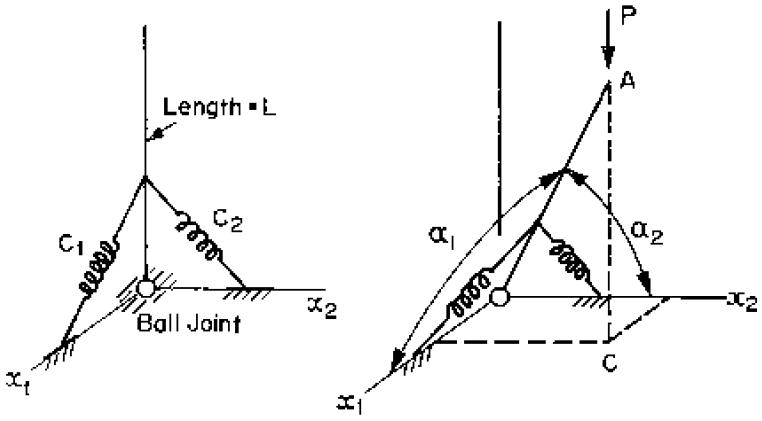


Figure 22. Model used by Croll and Walker (1972) to illustrate interaction of critical loads

By introducing the potential energy of the illustrated model and deriving the equilibrium equations four solutions are obtained, the next figures depict their expression:

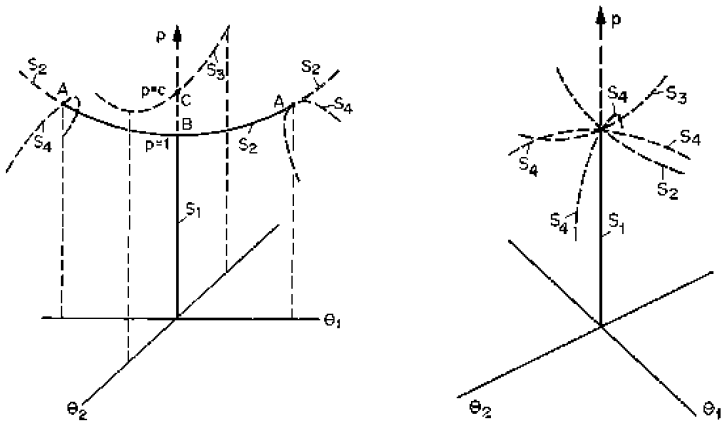


Figure 23. a) Stability paths for coupling between relatively close critical loads; b) Stability paths for coupling between coincident critical loads

It may be observed that the coupling of two stable symmetric post-critical curves (S2 and S3 in Figure 23) generates a fourth path (S4) which inverted shape suggests a marked increase in sensitivity of this model to initial imperfections.

Considering only the lowest post-critical path (S2) would have carried to have not detected the presence of these unstable paths, thus leading to an incorrect interpretation of the member behavior.

- Interaction Types in Bar Members

The characteristic of main interest of coupled instabilities is the evaluation of the associated maximum load reduction; this fact is generated by a column increase of sensitivity to geometric imperfection. As for simple buckling, there are coupled instabilities for which the effects of imperfections are not very important, but structures exist with very important reduction in the buckling loads.

If the two modes that couple have wavelengths of about the same order the post-critical path shows a weak or moderate interaction: this is the case of coupling between flexural and flexural-torsional buckling of mono-symmetrical compressed members.

If high difference exists between the two buckling modes wavelengths, as for an overall buckling that couples with a local one, a big erosion might occur.

For thin-walled members, a typical situation is a preliminary coupling between multiple local buckling modes with m-1, m, m+1 half-wavelength which give rise to an unstable post-critical path characterized by the *localisation of the buckling patterns* (experimental tests showed that the periodical local buckling mode develops at failure a local mechanism). The second interaction occurs between a stable post-critical overall buckling mode (i.e. flexural or torsional-flexural) with the unstable localized mode, this leads to a very unstable post-critical behavior.

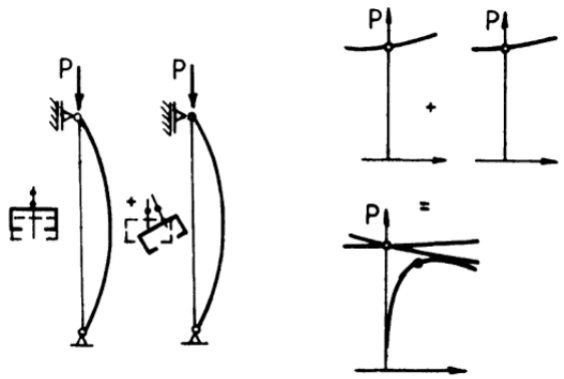


Figure 24. Flexural and Flexural-torsional coupled instabilities

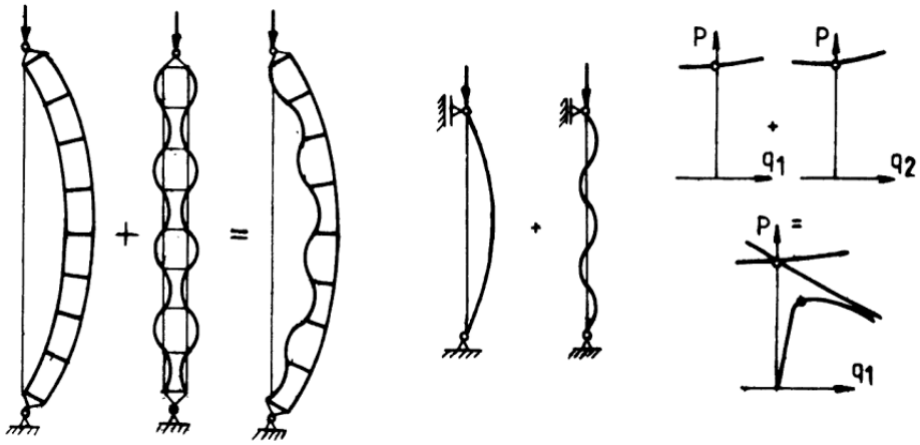


Figure 25. Overall and localized coupled instabilities

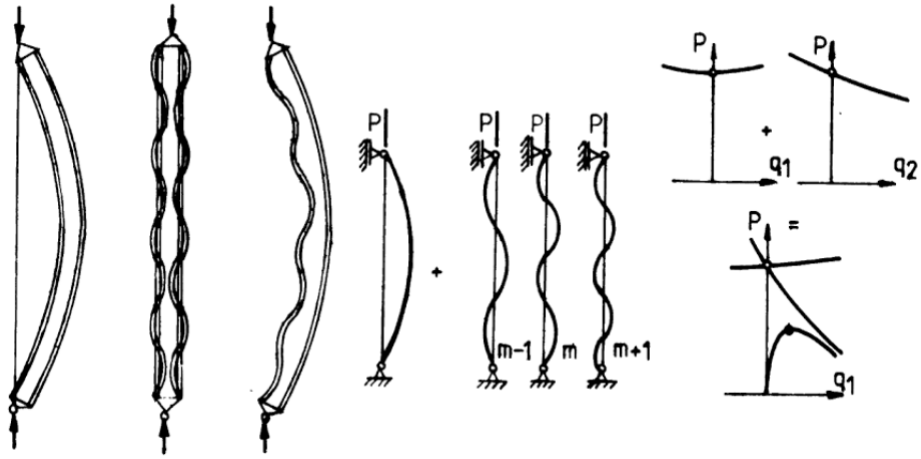


Figure 26. Multiple buckling mode interaction

Figure 24 – 26 summarize the main results achieved in the field.

Dubina [**26**] suggested a reduction from the theoretical buckling load less than 30% in the coupling between flexural and flexural-torsional buckling, more than 50% in the coupling between an overall mode with localized one.

Thus it might be concluded that coupling instabilities may generate high sensitivity to local and global geometric imperfections, this phenomenon will occur with a more and more significant relevance when approaching the simultaneous buckling mode.

For the studied cross section, doubly symmetric cross section, for which centroid and shear center coincide, these buckling modes are independent. In this case the column buckles at the lowest of the critical loads associated with these modes.

In the case of asymmetric cross sections, the modes are coupled and the lowest bucking load is lower than each one associated to the independent modes described above: in this case the column buckles in a flexural-torsional mode.

In the case cross section is mono-symmetric, in this case the flexural buckling mode about one axis is independent from the others modes because the shift between shear center and centroid with respect to this axis is zero. The shift of another axis on the other hand different from zero, this causes a flexural-torsional mode (deflection due to translation about non-zero shift axis and rotation about the shear center).

2.3.4 Open section, free and prevented warping

The theory of bending and torsion of thin-walled elastic column is applied in this part to solve the behaviour of a biaxially loaded column. The governing equation for column buckling, referring to Murray are achieved considering the equilibrium of a generic section which undergoes generic three *dof* small displacement (u, v, and φ).

The basic assumptions taken for this analysis are:

a. The material elasticity assurances the invariability of the sectional property
b. Thin-walled open cross section
c. Shear deformation is ignored
d. Initially straight member. The effects of initial lateral deflections and twist on the behavior of the beam-column can be neglected if they compared with the eccentricities applied
e. Small deformation. The lateral deflection *u*, *v*, and twisting θ are considered small. In this mode, it is possible to use linear differential equations
f. No lateral load, which is applied only in the beam-column ends.

The rectangular coordinate system (*x*, *y*, *z*) are defined in accordance with the principal axes of the cross-section having the origin at the mass center G, but the displacements (*u*, *v*, θ) are taken to shear center S of coordinates (x_o, y_o).

The equilibrium equations are as follows:

$$EJ_y u^{IV} + Pu^{II} - P\left(e_y - y_o\right)\vartheta^{II} = 0$$

$$EJ_x v^{IV} + Pv^{II} + P(e_x - x_o)\vartheta^{II} = 0$$

$$EJ_\omega \vartheta^{IV} + (K - GJ_T)\vartheta^{II} + P\left[(e_x - x_o)\vartheta^{IV} - \left(e_y - y_o\right)u^{II}\right] = 0$$

where:

$$K = \left[\frac{J_x + J_y}{A} + ({x_o}^2 + {y_o}^2)\right]P + \beta_x P e_x - \beta_y P e_y + \beta_\omega M_\omega$$

and β_x, β_y, β_ω are the Wagner's coefficients, that represent the symmetry degree of the section respect the relative axis. They are equal to:

$$\beta_x = \frac{1}{J_x}\int_A (x^2 + y^2)y dA - 2y_o$$

$$\beta_y = \frac{1}{J_y}\int_A (x^2 + y^2)x dA - 2x_o = 0$$

$$\beta_\omega = \frac{1}{J_\omega}\int_A (x^2 + y^2)\omega dA$$

This is a system of three linear non-homogenous differential equation with the unknown s*u*, *v* and θ. An exact solution is possible and has been written by *Culver*, who found the unknown variables resolving separately the homogenous particular problem. The twelve integration constants can be determined giving the followed twelve boundary conditions for both the extremes:

$$u = 0\ ;\ v = 0\ ;\ \theta = 0 \qquad \textit{geometric boundary conditions}$$

$$u^{II} = Pe_x/EJ_y\ ;\ v^{II} = Pe_y/EJ_x \qquad \textit{beam simple supported}$$

$$\theta^{II} = 0\ ;\ \theta^{I} = 0 \qquad \textit{for warping free or prevented}$$

In the exact approach, the evaluation of the integration constant it is not always simple and fast. There is an approximate way for find the same solution, and that studied by *Pekoz* and *Celebi* using the *Galerkin method*. They supposed the rotation and deflection functions with the following form:

$$u = A_0 u_0 - \frac{P e_x}{2EJ_y} z(L - z)$$

$$v = B_0 v_0 - \frac{P e_y}{2EJ_x} z(L - z)$$

$$\theta = C_0 \theta_0$$

Where u_0, v_0, θ_0 are function of z, and they respect the boundary conditions imposed, therefore simple supported beam-column.

Applying the *Galerkin method* to the previous differential equations, and using those displacements functions, a system of three linear equations with the unknowns A_0, B_0, C_0 is possible to be solved.

The solution of that system is the following matrix system:

$$\begin{bmatrix} P_y & 0 & P(e_y - y_0) \\ 0 & P_x - P & -P(e_x - x_0)K_1 \\ P(e_y - y_0) & -P(e_x - x_0)K_1 & {r_0}^2(P_z - P) \end{bmatrix} \begin{Bmatrix} A_0 \\ B_0 \\ C_0 \end{Bmatrix} = \begin{Bmatrix} -\frac{P^2}{P_y} e_x \\ -\frac{P^2}{P_x} e_y \\ P^2 \left[\frac{(e_y - y_0)e_x}{P_y} - \frac{(e_x - x_0)e_y}{P_x} \right] \end{Bmatrix}$$

where:

$$P_x = \frac{\pi^2 EJ_x}{L^2} flexural\ buckling\ load\ around\ x - axis$$

$$P_y = \frac{\pi^2 EJ_y}{L^2} flexural\ buckling\ load\ around\ y - axis$$

$$P_z = \frac{1}{{r_0}^2} \left(\frac{\pi^2 EJ_\omega}{L^2} K_2 + GJ_T \right) torsional\ buckling\ load$$

And the constants K_1 and K_2 are the integration coefficients functions of the boundary conditions used, which for this case take the following value 0.8834 and 4.1223 respectively in accordance with the *Pekoz* and *Celebi* solution.

In the case of mono-symmetry cross-section, which is subject to an axial force P in the plan of symmetry ($e_y = 0$) it is possible to write an independent equation of in-plane bending about x-axis and two homogenous simultaneous equations of flexural torsional buckling.

$$(P_y - P)A_0 = -\frac{P^2}{P_y} e_x$$

$$\begin{bmatrix} P_x - P & -P(e_x - x_0)K_1 \\ -P(e_x - x_0)K_1 & {r_0}^2(P_z - P) \end{bmatrix} \begin{Bmatrix} B_0 \\ C_0 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}$$

From which three critical loads are obtained:

$$P_{cr1,2} = \frac{(P_x - P_z) \pm \sqrt{(P_x - P_z) + 4P_xP_z((e_x - x_0)/r_0)^2{K_1}^2}}{2[1 - ((e_x - x_0)/r_0)^2{K_1}^2]}$$

$$P_{cr3} = P_y$$

2.3.5 Closed section, free warping

The theory explained in the previous paragraph can be used in case of closed section only in free warping, where the displacement along the profile are free.

The studied cross section is doubly-symmetrical, in this case the flexural buckling mode about *x* and *y*-axis is independent from other modes because the shift x_o and y_o between shear center and centroid is zero. Further, this cross-section is not susceptible to torsion due to coincidence of shear center and centroid.

2.3.6 Buckling of Thin Plates according to Eurocodes

The resistance of a thin plate under compressive forces is calculated in two steps; first calculating the critical load (elastic or buckling analysis) and second the determination of the ultimate load level (post-buckling analysis), when the behaviour is not linear any more due to plasticity or loss of stability.

EN-1993 part 1-5[**10**]defined the critical compressive stress for buckling of a plate element as:

$$\sigma_{cr} = k_\sigma . \frac{\pi^2 . E}{12.(1 - v^2)} . \frac{t^2}{b^2}$$

where; k_σ is the plate buckling coefficient that depends on the support conditions of the plate.

When an element is formed from a folded plate the width of the element b can be defined in different ways. EN 1993 part 1-5 does not take the bent parts of the cross-section into consideration and the use of EN 1993 part 1-6 is recommended by the standard. This should be done by calculating a notional flat width b_p of the plane elements, which is measured from the midpoints of the adjacent corner elements (Figure 27). Hence the critical stress becomes:

$$\sigma_{cr} = k_\sigma . \frac{\pi^2 . E}{12.(1 - v^2)} . \frac{t^2}{{b_p}^2}$$

Where; k_σ is the buckling coefficient (4 for internal compression elements and 0.43 for outstand compression elements).

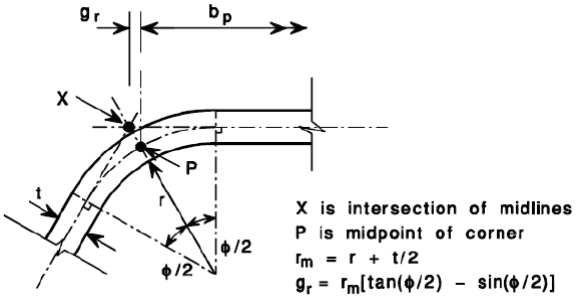


Figure 27. Notional flat width b_p

The minimum critical load is obtained from the outer plate both in using the elastic plate theory and in calculations with the notional flat width b_p.

- **The ultimate load for plates in compression**

In a linear elastic analysis of compression, the stress distribution is assumed to remain uniform until the plate buckles. In post-buckling however, stresses are re-distributed in the plate. Equilibrium paths, the influence of imperfections and load eccentricities may be analysed by means of equations of equilibrium or energy methods. In order to avoid complex analysis in design, the criteria of effective width is used, where a simply supported plate is considered and axially loaded in two sides, the stress distribution in a buckled cross-section of the plate is simplified into two stress blocks with constant stress over the total width. EN 1993 part 1-5 determined the effective width by using reduction factor ρ, this factor is obtained as follows:

$$b_{eff} = \rho . b$$

If the flat width b is replaced by the notional flat width b_p from EN 1993 part 1-3, equation above becomes:

$$b_{eff} = \rho . b_p$$

where;

$\rho = 1 \quad if \quad \overline{\lambda_p} \leq 0.673;$

$$\rho = \frac{1 - 0.055.(3 + \psi)}{\overline{\lambda_p}^2} \quad if \quad \overline{\lambda_p} > 0.673 \quad but \ \rho \leq 1.0$$

for doubly supported elements compression, where:

$\psi = \frac{\sigma_2}{\sigma_1}$is the ratio of the extreme fibre stresses;

$$\rho = \frac{1 - 0.188}{\overline{\lambda_p}^2} \qquad but \ \rho \leq 1.0$$

for outstand compression element.

The reduction factor ρ depends on the plate slenderness $\overline{\lambda_p}$ which is determinate in EN 1993 part 1-3 as:

$$\overline{\lambda_p} = \sqrt{\frac{f_y}{\sigma_{crit}}} = \frac{\frac{b_p}{t}}{28.4\,\varepsilon\sqrt{k_\sigma}}$$

If we compare the notional flat width bp and the effective width b_{eff} we get two expressions for the cross section area as:

$$A_g = \sum_{i=1}^{n} b_p \,.\, t$$

$$A_{eff} = \sum_{i=1}^{n} b_{eff} \,.\, t$$

In the ultimate limit state EN 1993-1-3 differs from part 1-5 since it allows a consideration of an average yield strength f_{ya} (due to different levels of cold working) if there is no reduction due to local and/or distortional buckling, this average yield strength f_{ya} is illustrated in equation 15 and it is found in EN1993 part 1-3 chapter 6.1.3(1) [**9**]. Thus the characteristic resistance for uniform compressed plate should be determined as follows:

$N_{c,RK} = A_{eff}.f_{yb}$ if $A_{eff} < A_g$

$N_{c,RK} = A_g\left[f_{yb} + \left(f_{ya} - f_{yb}\right).4.\left(1 - \frac{\lambda}{\lambda_{el}}\right)\right] \leq A_g.f_{ya}$ if $A_{eff} = A_g$

Due to local buckling the centroid of the effective cross-section does not coincide with the centroid of the gross cross-section described in EN1993 part 1-5 chapter 4.3(3) [**10**]. Therefore, an additional bending moment, considering the shift of the centroidal axis, should be considered. The characteristic buckling resistance for flexural buckling of a column made of thin plate is based on the relative slenderness $\overline{\lambda_p}$:

$\bar{\lambda} = \sqrt{\frac{A.f_y}{N_{cr}}}$ for class cross-section 1, 2, 3

$\bar{\lambda} = \sqrt{\frac{A.f_y}{N_{cr}}} \equiv \frac{L_{cr}}{i}\frac{\sqrt{\frac{A_{eff}}{A}}}{\lambda_1}$ for class cross-sections 4

$\bar{\lambda} = \pi\sqrt{\frac{E}{f_y}}$

Using the relative slenderness $\bar{\lambda}$ and an imperfection factor α of 0.49 (in accordance with buckling curve c in EN 1993 part 1-1) the reduction factor χ for the buckling resistance can be calculated as:

$$\chi = \frac{1}{\phi + \left[\phi^2 - \bar{\lambda}^2\right]^{0.5}}$$

$$\phi = 0.5.\left(1 + \alpha\left(\bar{\lambda} - 0.2\right) + \bar{\lambda}^2\right)$$

This leads to a characteristic buckling column resistance of:

$N_{b,Rk} = \chi . A . f_y$ for class cross-section 1, 2, 3

$N_{b,Rk} = \chi . A_{eff} . f_y$ for class cross-section 4

2.3.7 Buckling of cold-formed sections according to Eurocodes

- **Geometrical Proportions. EC3-1-3 -5.2**

The part considers the slenderness of the single plate, which composes the cross-section. The maximum ratio *b/t* have to be lower of a particular value, which depend on the relative position in the cross-section of the elements tested and the presence of stiffeners as shown in table 5.1 of Eurocode 3 part 1-3.

The maximum width-to-thickness ratios for the section studied are satisfied in all cases, with all the thickness.

- **Local and Distortional Buckling. EC3-1-3 -5.5**

In this part is considered the local and distortional buckling, that influence is accounted through the definition of effective cross section properties. This means that the non-uniform stress distribution that arise in the post local (or distortional) buckling range, is replaced by an uniform distribution of the maximum stress acting on a reduced portion of the element, having same thickness but reduced width.

The researches of the effective reaction section applied at an entire global section is complicated, and for that reason the assumption of the Eurocode and other codes, to consider the member as an assembly of individual plates simply supporting each other at boundaries. In the studied cross-section, every side is considered simply supported, included the flange and its stiffener, which dimension is not so important to influence the effective area of the flange.

The effective width of a compression element should be based on the comprehensive stress $\sigma_{com,Ed}$ in the element when the cross-section resistance is reached. Since this value is not initially known, an iterative procedure needs to be performed until the initial stress used to calculate the effective area coincides with the stress corresponding to the cross section resistance.

Local buckling can occur either simultaneously with distortional buckling, or at higher or lower loads. The two modes can interact too but the post–critical coupled mode is stable and, consequently, the local and distortional buckling strengths can be assessed independently of whether they occur simultaneously. In design codes, these two problems are treated separately.

- **Plane Element with Edge or Intermediate Stiffener. EC3-1-3 -5.5.3**

For the studied case the element are classified as doubly supported compression element and outstand compression element.

The effective width b_{eff} of a compression element is defined as:

$$b_{eff} = \rho b$$

The reduction factor ρ is based on the largest compressive stress $\sigma_{com,Ed}$ acting on the element when the cross section resistance is reached. The calculation for taking into account local buckling is done in accordance to EC3-1-5, as described in section 3.3.6.

Distortional buckling of compression members is governed by the rotational stiffness at the web/flange junction; deeper webs are more flexible and thus provide less rotational stiffness to the web/flange juncture. EN1993-1-3 does not provide explicit provisions for distortional buckling. However, a calculation procedure can be obtained from the interpretation of the rules given in the code for plane elements with edge or intermediate stiffeners in compression.

The procedure is based on the assumption that the stiffener behaves as a compression member with continuous partial restraint. This restraint has a spring stiffness that depends on the boundary conditions and the flexural stiffness of the adjacent plane elements of the cross section. The spring stiffness of the stiffener may be determined by applying a unit load per unit length to the cross section at the location of the stiffener. The rotational spring stiffness C_θ characterizes the bending stiffness of the web part of the section. The spring stiffness K per unit length may be determined from:

$$K = u/\delta$$

where δ is the deflection of the stiffener due to the unit load u.

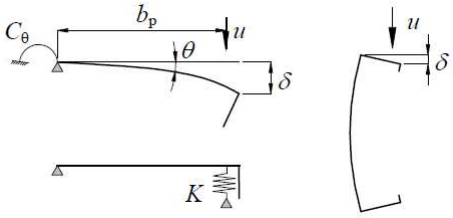


Figure 28. Procedure for spring stiffness K according to EN1993-3

The elastic critical buckling stress for a long strut on an elastic foundation, in which the preferred wavelength is free to develop, is given by Timoshenko & Gere (1961):

$$\sigma_{cr} = \frac{\pi^2 . E . I_s}{A_s . \lambda^2} + \frac{I}{A_s . \pi^2} K . \lambda^2$$

where:
A_s and I_s are the effective cross sectional area and second moment of area of the stiffener according to EN1993-1-3, as illustrated in Figure 29 for an edge stiffener;

$\lambda = L / m$ is the half-wavelength; m is the number of half-wavelengths.

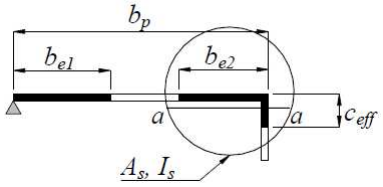


Figure 29. Effective cross sectional area of stiffener

The preferred half-wavelength of buckling for a long strut can be derived from equation above by minimizing the critical stress:

$$\lambda_{cr} = \sqrt[4]{\frac{E.I_s}{K}}$$

For an infinitely long strut, the critical buckling stress can be derived, after substitution, as:

$$\sigma_{cr} = \frac{2.\sqrt{K.E.I_s}}{A_s}$$

The above equation of critical buckling stress is given in EN1993-1-3; thus, this method does not consider the effect of column length but assumes that it is sufficiently long for integer half-waves to occur.

2.3.7.1 Ultimate Limit States. EC3-1-3 -6

The design value of the internal force and moment at each cross section shall not exceed the design value of the corresponding resistance, which will be determined for combined bending moment and axial compression.

- **Resistance of cross section. EC3-1-3 -6.1**

a. Axial Compression. EC3-1-3 -6.1.3

The design resistance for uniform compression f_{yb} / γ_{M0} is determined as the resistance of the effective area A_{eff}:

$$N_{c,Rd} = A_{eff} f_{yb} / \gamma_{M0}$$

b. Bending Moment. EC3-1-3 -6.1.4

The design moment resistance for bending around one principal axis $M_{c,Rd}$ is determined as follows:

$$M_{c,Rd} = W_{eff} f_{yb} / \gamma_{M0}$$

where W_{eff} is the effective resistance of the cross section and a linear distribution of stress across the cross-section is taken.

c. Combined Compression and Bending. EC3-1-3 -6.1.9

Cross-section subject to a combined axial force and bending moment should satisfy the following criteria:

$$\frac{N_{Ed}}{N_{c,Rd}} + \frac{M_{y,Ed} + \Delta M_{y,Ed}}{M_{cy,Rd,com}} \leq 1$$

where:

$\Delta M_{y,Ed} = N_{Ed} e_{Ny}$ is the additional moment for the eccentricity between the gross area and the effective one

$M_{cy,Rd,com}$ is the pure moment resistance

2.3.7.2 Buckling Resistance. EC3-1-3 -6.2

a. Axial Compression. EC3-1-3 -6.2.2

The design buckling resistance of a compression member should be taken as:

$$N_{b,Rd} = \frac{\chi N_{c,Rd} \gamma_{M0}}{\gamma_{M1}}$$

where:

$\chi = \frac{1}{\phi + (\phi^2 - \bar{\lambda}^2)^{0.5}}$ is the reduction factor for the relevant buckling mode,

$$\phi = 0.5[1 + \alpha(\bar{\lambda}^2 - 0.2) + \bar{\lambda}^2]$$

α is an imperfection factor, depending on the relevant buckling mode
$\bar{\lambda}$ is the relative slenderness for the relevant buckling mode

The appropriate imperfection factor for the relevant buckling mode should be obtained from Table 6.2 in EC3.

In part EC3-6.2.2.2 (3) is prescribed that for closed built-up section (as could be intended the studied one) curve b should be used which corresponding α is equal to 0.34.

a.1. Flexural Buckling. EC3-1-3 -6.2.2.2

The relative slenderness $\bar{\lambda}$ for flexural buckling about a given axis is determined as:

$$\bar{\lambda} = \frac{L_{cr}}{i} \frac{\sqrt{A_{eff}/A}}{\lambda_1}$$

with:

$$\lambda_1 = \pi \sqrt{\frac{E}{f_y}} = 93{,}9\varepsilon$$

L_{cr} is buckling length in the plane considered
i is radius of gyration based on the properties of the gross section

a.2. Torsional and Torsional-Flexural Buckling. EC3-1-3 -6.2.2.3

In this part is stated that all section should be verified for torsional or flexural torsional buckling if these modes result to be the critical ones.

The relative slenderness $\bar{\lambda}$ for the torsional-flexural buckling is determined as:

$$\bar{\lambda} = \sqrt{\frac{A_{eff}/A}{N_{cr}}}$$

Where N_{cr} is the axial critical load for the pertinent instability, therefore the minimum critical load between:

$N_{cr,T}$ elastic critical force for torsional buckling

$N_{cr,TF}$ elastic critical force for flexural-torsional buckling

For simply supported beam

$$N_{cr,T} = \frac{1}{{i_0}^2}(GJ_t + \frac{\pi^2 EJ_\omega}{l^2}$$

where

G shear modulus

J_t torsion constant of gross-section

J_ω warping constant of gross-section

$${i_o}^2 = {i_z}^2 + {i_y}^2 + {y_o}^2 + {z_o}^2$$

i_z radius of gyration of the gross-section about z-z axis

i_y radius of gyration of the gross-section about y-y axis

y_o, z_o coordinate of shear center respect to gross-section center

l buckling length

For mono-symmetric cross section

$$N_{cr,TF} = \frac{N_{cr,y}}{2\beta}[1 + \frac{N_{cr,T}}{N_{cr,y}} - \sqrt{\left(1 - \frac{N_{cr,T}}{N_{cr,y}}\right)^2 + 4\left(\frac{y_o}{i_o}\right)^2 \frac{N_{cr,T}}{N_{cr,y}}}]$$

with

$$\beta = 1 - \left(\frac{y_o}{i_o}\right)^2$$

b. Bending Moment. EC3-1-3 -6.2.3

b.1. Lateral torsional buckling

The design buckling resistance of a member susceptible to lateral torsional buckling should be taken as:

$$M_{b,Rd} = \frac{\chi_{LT} M_{c,Rd} \gamma_{M0}}{\gamma_{M1}}$$

where:

$$\chi_{LT} = \frac{1}{\phi_{LT} + [{\phi_{LT}}^2 - 0.75\overline{\lambda_{LT}}^2]} \leq 1$$

$$\phi = 0.5[1 + \alpha_{LT}\left(\overline{\lambda_{LT}}^2 - 0.4\right) + 0.75\overline{\lambda_{LT}}^2]$$

$$\overline{\lambda_{LT}} = \frac{\sqrt{M_{c,Rd}}}{M_{cr}}$$

α_{LT} = 0.34 is an imperfection factor

M_{cr} is the elastic critical moment of the gross-section for lateral-torsional buckling, which is taken in literature

The M_{cr} for mono-symmetric cross-section can be written as

$$M_{cr} = \sqrt{\frac{\pi^2 EJ_z}{l^2}} \sqrt{GJ_\omega + \frac{\pi^2 EJ_\omega}{l^2} + [\frac{\beta_y}{2}\sqrt{\frac{\pi^2 EJ_z}{l^2}}]^2} + \frac{\beta_y}{2}\sqrt{\frac{\pi^2 EJ_z}{l^2}}$$

where

β_y is the Wagner coefficient around the y-y axis, which represents the symmetry degree of the section.

c. Bending and Compression. EC3-1-3 -6.2.4

For member with mono-symmetrical cross section the resistance criteria used is:

$$\left(\frac{N_{Ed}}{N_{b,Rd}}\right)^{0.8} + \left(\frac{M_{Ed}}{M_{b,Rd}}\right)^{0.8} \leq 1$$

2.3.8 Buckling of shell sections according to Eurocodes

Thin cylindrical shells structures under compressive stresses are known to be sensitive to imperfections which reduce their resistance considerably. Therefore, local buckling is one of the most important criteria in the calculations of the maximum strength of such shells. The bending stress which varies trough the shell thickness, does not affect the stability of the shell, except in as much as they may cause yield of the material of the shell and in this case local reduction in the stiffness of the shell. In general there are two ways in which an elastic structure may become unstable. These are commonly termed as snap-through and bifurcation point illustrated in Figure 18.

a. Elastic buckling

For thin circular shells the elastic theory of the shell is given by Timoshenko where the elastic critical stress in the case for symmetrical buckling of the thin shell is defined as:

$$\sigma_{crit} = \frac{1}{\sqrt{3.(1-\upsilon^2)}}.\frac{E.t}{r} \cong 0.605\,.\frac{E.t}{r}$$

where, r is the radius of the shell and t is the wall thickness.

When analysing buckling in a shell, it is important to consider in addition to axial compression, the strain of the middle surface in the circumferential direction. Thus the strain energy of the shell is increased. This increase in energy must be equal to the work done by the compressive load as the cylinder shortens owing to buckling. The buckling mode can be given as:

$$w = w_o.sin\left(\frac{m.\pi.x}{l}\right).\sin(n.\varphi)$$

The critical load of a structure in the elastic range can be obtained from eigenvalue analysis if the geometry of the structure up to buckling remains essentially unchanged as in a compressed column. However, in many practically important buckling problems for shells e.g. for axis symmetrically loaded shells of revolution the critical load and the buckling mode is significantly affected by pre-buckling geometric nonlinearity. Linear eigen value analysis is required to find the elastic bifurcation point on the pre-buckling path and thus predict the critical load and mode.

b. Ultimate resistance of thin shell

Shell buckling is the principal design consideration for cylindrical shells with constant wall thickness under any stress loading conditions because the wall thickness of the shell may vary. Therefore, buckling stress becomes an important concept for the assessments of cylindrical shells. Buckling stress design approach follows the principle established in Eurocode. The calculation process of the ultimate strength for cylindrical shells begins with the determination of two important variables:

1. The plastic limit load
2. The linear elastic critical load

The ratio of these two loads is used to determine the shell relative slenderness $\bar{\lambda}$ of the shell:

$$\bar{\lambda} = \sqrt{\frac{f_y}{\sigma_{crit}}}$$

Where f_y is the yield strength of the shell and $\sigma_{crit}$is the critical stress value of the shell. The combination of the relative slenderness with different reduction factors governs the assessment of the relative plastic and elastic behaviour of the shell.

The buckling stress calculation is done based on the evaluation of the ultimate load according to the EN 1993 part 1-6 and EN 1993 part 3-2 which are defined by the variation of the reduction factors of:

1. Geometry
2. Load case
3. Fabrication quality

The critical stress in EN 1993 part 1-6 is determined by the variation of Donnell's theory with the introduction of the coefficient factor C_x. The factor C_x refers to the length of the shell defined from long to short cylinders where:

$C_x = 1$ for medium-length cylinders if $1.7 \leq \omega \leq 0.5 . \frac{r}{t}$

$C_x = 1.36 - \frac{1.83}{\omega} + \frac{2.07}{\omega^2}$ for short cylinders if $\omega \leq 1.7$

$C_x = C_{x,N}$ for long cylinders if $\omega > 0.5 . \frac{r}{t}$

$$C_{x,N} = 1.36 - \frac{0.2}{C_{x,b}}\left[1 - 2.\omega.\frac{t}{r}\right]$$

$C_{x,N}$ represents the reduction in the elastic critical stress. And $C_{x,b}$ is a parameter depending on the boundary conditions. The idea to classify shells into strength groups according to the quality of construction was introduced into the Eurocode. The characteristic determinate factor used to calculate the sensitivity of the elastic buckling strength of the shell to both imperfections and geometry non-linearity is calculated as follow:

$$\Delta w = \frac{1}{Q}.\sqrt{\frac{r}{t}}.t$$

Where Q is the meridional compression fabrication quality parameter with three fabrication tolerance quality class. The meridional buckling parameter factor α_x is determined as a function of the non-dimensional imperfection amplitude $\Delta w/t$ since this measure raised in all imperfection sensitivity analyses for the geometric and quality fabrication of the shells.

$$\alpha_x = \frac{0.62}{1 + 1.91\,.\left(\frac{\Delta w}{t}\right)^{1.44}}$$

The buckling reduction factor χ is determined as a function of the relative slenderness of the shell $\bar{\lambda}_x$ from:

$\chi = 1$ if $\bar{\lambda}_x \le \bar{\lambda}_0$

$\chi = 1 - \beta.\left(\frac{\bar{\lambda}_x - \bar{\lambda}_0}{\bar{\lambda}_p - \bar{\lambda}_0}\right)^{\eta}$ if $\bar{\lambda}_0 < \bar{\lambda}_x < \bar{\lambda}_0$

$\beta = 0.6$ and $\eta = 1.0$

$\chi = \left(\frac{\alpha_x}{\bar{\lambda}_x{}^2}\right)$ if $\bar{\lambda}_p \le \bar{\lambda}_x$

where:

β is the plastic range factor

η is the interaction exponent

$\bar{\lambda}_0$ is the meridional squash limit for short and medium cylinder is defined as:

$$\bar{\lambda}_0 = 0.2 + 0.1\left(\frac{\sigma_{x,M}}{\sigma_x}\right)$$

And $\bar{\lambda}_p$ is the plastic relative slenderness, defined as:

$\bar{\lambda}_p = \sqrt{\frac{\alpha_x}{1-\beta}}$ The plastic relative slenderness

Hence, the characteristic buckling strength is:

$$\sigma_u = \chi\,.f_{yd}$$

This procedure of shell assessment will be used in Chapter 4 when calculating the ultimate resistance of members. Diagram below shows the workflow of the of the shell assessment with the above mentioned methods.

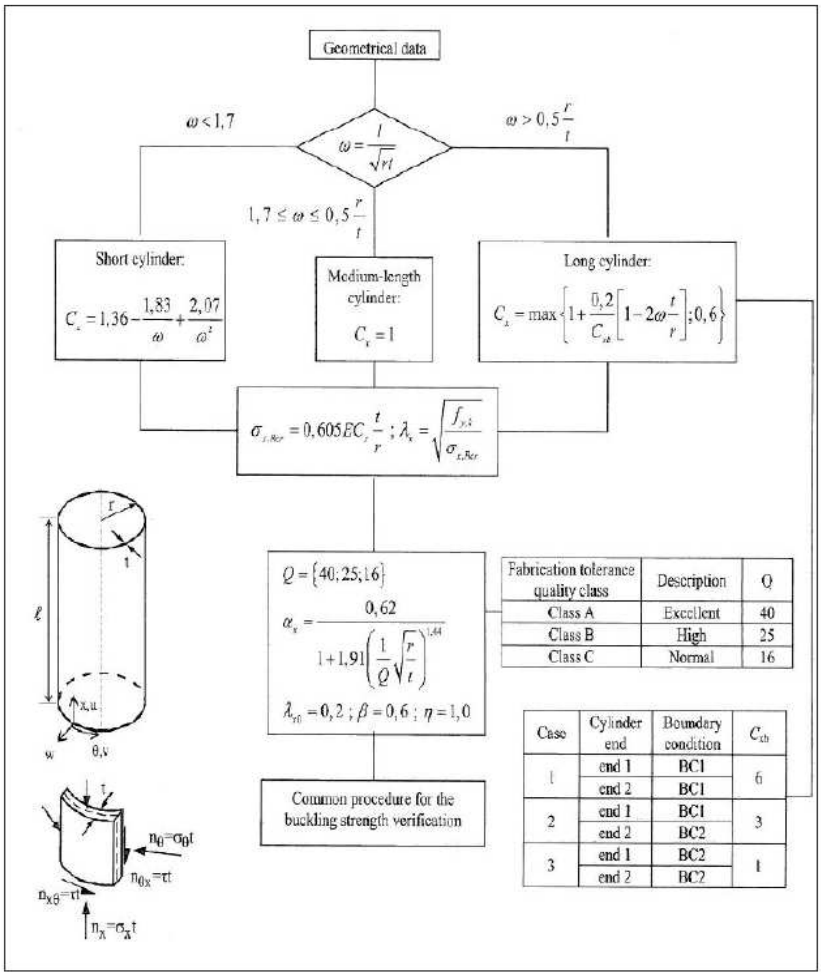


Figure 30. Workflow of resistance analysis of cylindrical shell based on EN 1993 part 1-6

2.4 Connection in Lattice(Truss) Structures

So far the focus of all the research carried out on bolted connections in steel structures has been on the strength of such joints. However it has long been realized that, as well as strength, connection stiffness is important to a sound design and successful use of steel in any building. Traditionally, connections have been designed as either pinned (implying no moment transfer) or rigid (implying complete rotational continuity) in steel frames - alternatively referred to as simple and continuous construction. Lattice structure is no exception to this tradition. At present, the truss design is constantly being carried out with ideally hinged joints in the finite element models.

The notions of pinned and rigid joints are simply extreme cases of true joint behavior. Most pinned connections possess some rotational stiffness, while rigid connections often display some flexibility. In real systems, it would therefore seem more appropriate to regard all steel frames under the more general heading of semi-rigid construction, treating the simple and continuous construction as the extremes (Figure 30). The so-called semi-rigid joint covers the whole range of intermediate characteristics between the two idealized ones mentioned above. In beam analysis, the performance of such connections can be represented by set of springs with respective degree of freedoms and stiffness values. The structural design code [**27**] provides guidelines for engineers, however they do not widely

apply probably due to difficulties of assessment and/or insufficient guidelines of such connections. Finite element analysis is then needed to know the actual stiffness of the connection.

This thesis is made to describe a finite element investigation on rotational stiffness of semi-closed polygonal steel section members in lattice structures. In practice, the stiffness of the connection obtained from moment-rotation relation would represent the real rigidity of that connection, which affects the behavior of structure. The stiffness of connection will affects distribution of both internal forces and moments as well as deformations of the structure. Moreover, it affects the buckling length factor which in turn determines the first order elastic buckling load of a member. The first order buckling load, material properties and a buckling curve are then used to determine an elastic-plastic buckling load. In many cases, the significance of connection stiffness to the response of entire structure is not negligible [**28**]. This shows the importance of determination of the real stiffness of connections rather than assuming it as fully hinge (pinned) or fixed (rigid) ones.

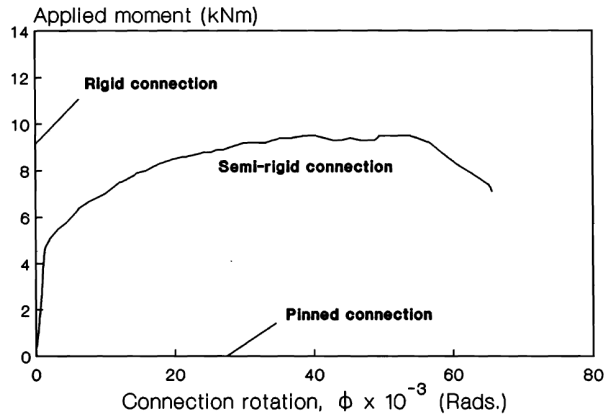


Figure 31.Typical moment-rotation characteristics of a bolted connection

Figure 32 shows the bending moment distributions for a pinned, rigid and semi-rigid beam. It can be seen that with pinned connections the mid-span moment is critical, whereas with rigid joints the end-moments are critical. If semi-rigid joints are used, these two moments may be more nearly balanced, and the optimum solution is when the mid-span moment and the end-moments are equal. Hence, consideration of the moment/rotation characteristics of bolted connections can be of vital importance to economic design.

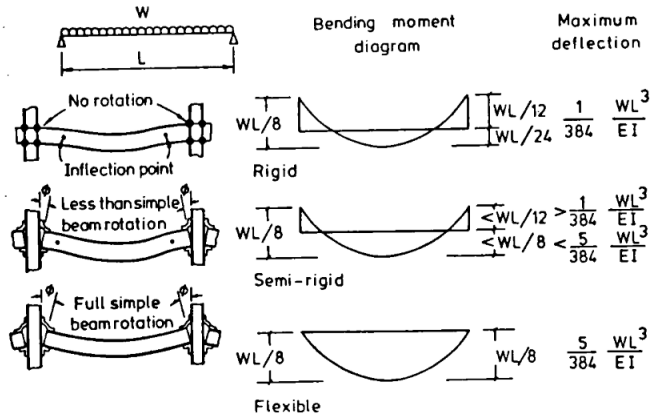


Figure 32. Effect of end restraint on end moments and deflections for elastic response

The thesis consists of two parts; a literature study and a report about the analytical and numerical investigation. The first part is made to gain insight into the moment-rotation characteristics of members in the lattice girder, with emphasis on hollow structural members in lattice structures, and to make clear what has already been done and what is known. The second part describes the performed numerical study, where the methodology, used parameters and conclusions are discussed with emphasis on connections with semi-closed polygonal sections. This novel type of section is a new alternative in steel lattice girder structures so that not many researches have been performed, especially related to its connection in which this thesis tries to explore.

2.4.1 Moment-Rotation Behaviour

The non-linear moment-rotation (M-φ) behavior of a semi-rigid joint can be idealized as a series of stepwise linear relationships of gradient R_1, R_2, ..., R_n, as shown in Figure 33. If at a general loading increment j, the angle of rotation of the semi-rigid joint changes from φ_{j-1}, to φ_j caused by change in rotational moment from M_{j-1} to M_j; then the incremental moment-rotation of the joint may be defined as:

$$\delta M = R_j . d\varphi$$

where,

$$R_j = (M_j - M_{j-1})/(\varphi_j - \varphi_{j-1})$$

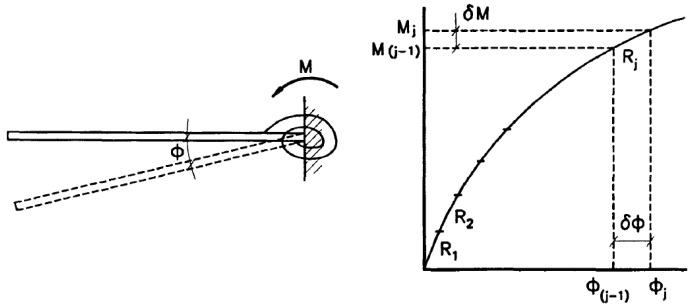


Figure 33. Moment-rotation behaviour of a semi-rigid joint

The hinge can be replaced with a rotational spring to model the semi-rigid nature of the joint. To consider the appropriate stiffness of the spring a quantity called fixity factor is introduced. The fixity factor is defined as ratio of angles seen Figure 34.

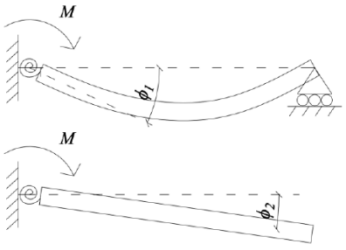


Figure 34. Definition of fixity factor

It can be written as equation (Monforton and Wu, 1963)

$$\alpha = \frac{\varphi_2}{\varphi_1} = \frac{\frac{M}{k+\frac{EI}{L}}}{\frac{M}{k}} = \frac{1}{1+\frac{3EI}{kL}}$$

or if fixity factor is known, the respective rotational stiffness can be solved, thus

$$k = \frac{3EI}{L}\frac{\alpha}{1-\alpha}$$

Fixity factor can get values $\alpha \in [0,1]$ where 0 means ideally hinged and 1 means ideally rigid connection. Eurocode 1993-1-8[**27**], on the other hand, gives values for the limits of ideally rigid or hinged joints. A joint can treated ideally rigid if the stiffness satisfies

$$k \geq \frac{25EI}{L}$$

or by using fixity factor

$$\alpha \geq \frac{25}{28} = 0.893$$

Also, if stiffness satisfies

$$k \leq \frac{EI}{2L} \quad or \quad \alpha \leq \frac{1}{7} = 0.143$$

the joint can be treated ideally hinged.

2.4.2 Rotational stiffness based on Eurocode 3-1-8

EN 1993-1-8 [**27**] distinguishes between three simplified joint models: A simple model in which the joint is assumed to be nominally pinned, thus not transmitting bending moments; a continuous joint model in which the joint behaviour is considered not to have any effect on the analysis; and a semi-continuous model in which the behaviour of the joint has to be taken into account in the global analysis. Three different kinds of global analyses are considered. In an elastic global analysis a joint is classified according to its rotational stiffness, whereas in a rigid-plastic analysis it is categorised based on its strength. An elastic-plastic global analysis requires both strength and stiffness for classification. A short summary is depicted in Table 3.

Table 3. Different joint models according to EN 1993-1-8 [27]

Method of global analysis	Classification of joint		
Elastic	Nominally pinned	Rigid	Semi-rigid
Rigid-Plastic	Nominally pinned	Full-strength	Partial-strength
Elastic-Plastic	Nominally pinned	Rigid and full-strength	Semi-rigid and partial-strength Semi-rigid and full-strength Rigid and partial-strength
Type of joint model	Simple	Continuous	Semi-continuous

Joints which are classified according to their design moment resistance are divided into three classes – nominally pinned, partial-strength and full-strength. Nominally pinned joints are defined as joints with a design moment resistance $M_{j,Rd}$ smaller than 0.25 times the design moment resistance of a full-strength joint. In order to classify a full-strength joint a comparison of its design moment resistance $M_{j,Rd}$ is made with respect to the design plastic bending moment resistance $M_{pl,Rd}$ of the adjacent members (beam or column). All joints which do not meet the criteria for nominally pinned or full-strength joints are considered to be partial-strength.

If a joint is classified by its stiffness, it should be categorized into nominally pinned, rigid and semi-rigid joints. Nominally pinned joints shall transmit internal forces without developing significant moments, whereas rigid joints are assumed to have sufficient rotational stiffness as to fully transfer the moment acting on the connection. All joints which do not meet the criteria for nominally pinned or rigid joints are considered to be semi-rigid.

When the connection is classified as semi-rigid, the flexibility must be determined. The flexibility is classified as an initial stiffness ($S_{j,ini}$) and a stiffness (S_j). If the moment in the joint is smaller than 2/3 of the maximum moment capacity of the joint ($M_{j,Ed} < 2/3\ M_{j,Rd}$), the initial stiffness $S_{j,ini}$ of the joint can be used. If larger moments can occur (not greater than the maximum moment capacity of the joint), the stiffness S_j must be used. As a simplification, $S_{j,ini}$ / η may be used for any value of the moment in the joint (not greater than the maximum) where η is the stiffness modification factor, which is 2 for girder-column connections and 3 for all other connections if the connection is bolted end-plates (Figure 35).

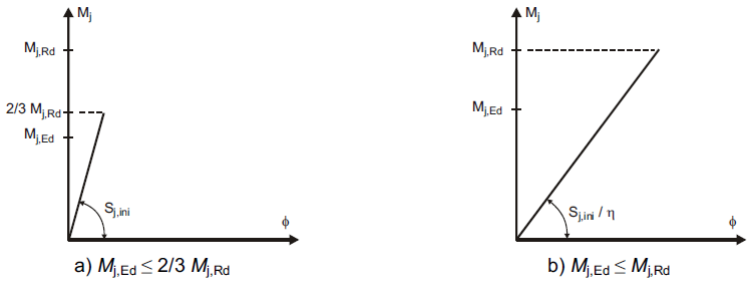


Figure 35. Rigidity in elastic calculation (Eurocode 3:1-8) [27]

The determination of the stiffness is not discussed for bolted connections of lattice systems. For these connections, only the capacity (strength) of the connections is considered, with assumption of pinned joints.

2.5 Stiffness of Lip Connection

The structure with bolted joints to be analyzed is discretized with a number of elements and then assembled at nodes. The elements of different type and shape with complex loads and boundary conditions can be used simultaneously using FEM. Consider an element of volume V bounded by a surface S with the traction vector $\bar{\mathbf{t}}$ prescribed on a part of the surface S_F. The finite element formulation is to begin with a variational principle related to total potential energy as follows:

$$\pi = \int_V \boldsymbol{\sigma}^{\mathrm{T}} \varepsilon dV - \int_{S_F} \mathbf{u}^{\mathrm{T}} \bar{\mathbf{t}}\, dS = 0,$$

where σ, $\boldsymbol{\varepsilon}$ and **u** are stress, strain and displacement vector, respectively. The first order variation of the functional equation above can be written as

$$\delta\pi = \int_V \boldsymbol{\sigma}^{\mathrm{T}} \delta\varepsilon dV - \int_{S_F} \delta\mathbf{u}^{\mathrm{T}} \bar{\mathbf{t}}\, dS = 0$$

Using constitutive equation $\boldsymbol{\sigma} = \mathbf{D}\boldsymbol{\varepsilon}$and strain–displacement relation $\boldsymbol{\varepsilon} = \mathbf{B}\mathbf{u}$, the above equation is derived is derived as

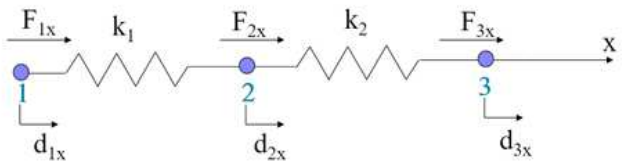
$$\delta\mathbf{u}^{\mathbf{T}} \left[\int_V \mathbf{B}^{\mathbf{T}} \mathbf{D} \mathbf{B} dV \right] \mathbf{u} - \delta\mathbf{u}^{\mathrm{T}} \int_{S_F} \mathbf{N}\bar{\mathbf{t}}\, dS = 0$$

where N is matrix of shape functions. Equation above is the basic equation for the finite element discretization and can be converted to algebraic equations as follows:

$$\mathbf{K}\mathbf{u} = \mathbf{f}$$

where **K** is the element stiffness matrix, **f** is the vector of surface loads. This equation system can be solved for unknown displacement vector **u** using commercial FEM software.

In this section definition of stiffness matrix of a structure is given. This concept will be used and developed in the next chapter for calculating stiffness of bolted connection along the lip of polygonal plates. Consider a simple example structure, as follows.



In general, the global stiffness matrix for this structure can be written:

$$\mathbf{K} = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix}$$

The finite element force-displacement relations:

$$\begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \\ d_3 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \\ F_3 \end{Bmatrix}$$

The first equation of the force equilibrium at node 1 is

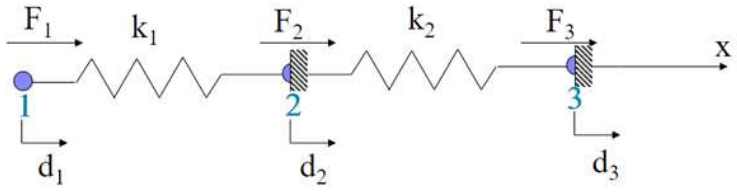
$$k_{11}d_1 + k_{12}d_2 + k_{13}d_3 = F_1$$

By applying displacement at node 1 while nodes 2 and 3 are held fixed, the force k_{21} and k_{31} can be obtained. Similarly, it allows knowing the other entries of the global stiffness matrix.

In general:

$\boldsymbol{k_{ij}}$ = Force at node 'i' due to **unit displacement** at node 'j' keeping **all the other nodes fixed**

This is an alternate route to generating the global stiffness matrix.



Taking into account the rotational degree of freedom, the above equation can be extended into:

$$\begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} & k_{15} & k_{16} \\ k_{21} & k_{22} & k_{23} & k_{24} & k_{25} & k_{26} \\ k_{31} & k_{32} & k_{33} & k_{34} & k_{35} & k_{36} \\ k_{41} & k_{42} & k_{43} & k_{44} & k_{45} & k_{46} \\ k_{51} & k_{52} & k_{53} & k_{54} & k_{55} & k_{56} \\ k_{61} & k_{62} & k_{63} & k_{64} & k_{65} & k_{66} \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ ur_1 \\ ur_2 \\ ur_3 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ M_1 \\ M_2 \\ M_3 \end{Bmatrix}$$

In the FEM modeling, each bolt connection on the lips provide translational (spring) stiffness in 2 DOF, i.e. *x*- and *y*-direction and rotational stiffness in 1 DOF, i.e. *z*-direction. Detail method and calculation of connection stiffness is given in Chapter 5.

2.6 Imperfections

In practice, a geometrically and materially perfect structure is non-existent. It is therefore essential to consider and include a proper imperfection into the numerical model. Due to the fact that the non-linear response of thin-walled structures are generally sensitive to imperfection, and consequently affects the overall strength of this kind of member, investigation was carried out with refer to the European standard in EN 1993 part 1-1 and EN 1993 part 1-5.

Imperfections of cold-formed steel members may include bowing, warping and twisting, also local deviations and bar deflections. As for the analysis in this study, global geometrical imperfection becomes the main issue. The magnitude of the imperfection in the member, depends of the shape of the buckling mode, which can be obtained from eigenbuckling analysis of the compressed member. Another imperfection, i.e. material imperfection which occurs in the member corners by cold-forming is residual stress, as already mentioned in previous section. It is complicated to adequately model residual stresses in the analysis. Lack of data makes selecting an appropriate magnitude difficult. As a result, residual stresses are often excluded from the analysis or the stress-strain behaviour of the material is modified to approximate the effect of residual stresses. In this thesis this effect was taken into account by using real properties of the material from coupon test experimental data.

2.7 Post-Buckling and Non-Linear FE Analysis

Thin-walled structures, in general, are slender structural elements rather sensitive for an influence of geometrical imperfections and eccentricity of applied load. In order to study the influence of these parameters on the behavior of the compressed member it is necessary to perform non-linear FE analysis.

Moreover, non-linear analysis is considered when non-linearity of the material, such as plasticity, and of the geometry of the profile, is present, or post-buckling behavior is of interest. This is the main objective of this section. This was possible by a step-by-step loading process, which simulates a more realistic behavior of the structure, which is programmed in ABAQUS/Standard. Incremental procedure based on RIKS algorithm is used to solve system of non-linear equations.

In linear elastic stress analysis, equilibrium is based on the original undeformed configuration; while for linear elastic instability problem, deformed shape is considered, although the deformation before instability is usually very small compared to structure's original geometry. For this type of problems, theory of linear elastic buckling analysis serves well in predicting the onsite of the buckling or critical loads. In other situations, when a structure undergoes finite deformation due to complex load or material plasticity before instability actually occurs, system parameters change along with the deformation, thus, makes the eigenvalue analysis inaccurate to perform. If the system accumulated deformation is not negligible prior to instability due to the loads that ultimately causes its instability, the critical load becomes system configuration or deformation dependent. In this case, a nonlinear analysis becomes necessary in order to simulate this type of highly unstable behavior due to lack of the inclusion of large deviation from the original geometry.

In nonlinear static analysis for buckling, post-buckling, or collapse behavior, the tangent stiffness from the load-displacement response curve could change signs when system changes its stability status as shown in Figure 36. The classical Newton's method will not work in this situation because the corrections for approaching equilibrium solutions during iterations may become difficult to determine when the tangent stiffness is close to null.

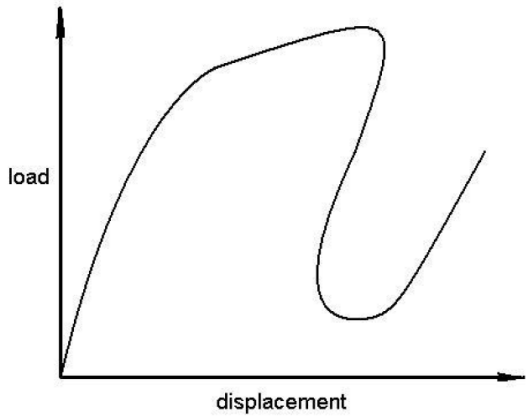


Figure 36. A typical unstable response curve

Static equilibrium states during the unstable phase of the response can be found by using the "modified Riks method" [**29**]. This method is used for cases where the loading is proportional; that is, where the load magnitudes are governed by a single scalar parameter. The basic Riks algorithm is essentially Newton's method with load magnitude as an additional unknown to solve simultaneously for loads and displacements, thus, can provide solutions even in cases of complex and unstable response such as that shown in Figure 36.

As for material plasticity, realistic description of the steel behavior is obtained from the coupon tests. Therefore, true stress-strain curve modified from the test results was used as an assumption for the material behaviour.

$$\sigma_{true} = \sigma(1 + \varepsilon)$$

$$\varepsilon_{true} = ln(1 + \varepsilon)$$

$$\varepsilon^{pl}{}_{true} = \varepsilon_{true} - \frac{\sigma_{true}}{E}$$

3 FINITE ELEMENT MODELLING OF THE PROPOSED STRUCTURAL MEMBERS

The application of cold-formed steel members for truss structure is relatively new in building construction. The knowledge of structural behaviour of this steel member especially for the proposed type of section is not widely acquired. Limitations due to little existing experience and research for this type of structure has prompted tests and experiments to be performed in order either to get better understanding of real behaviour of the structure or establish design approach. In this part an extensive parametric study by numerical FEM analysis was proposed. By doing so, it is expected that the structural characteristic and behaviour of semi-closed polygonal cross-section can be well-understood and bring in useful recommendation for design purpose. Parametric studies were carried out through automation using MATLAB and Python script. Scripting is a powerful tool for calculating a large number of cross-sections in parametric studies that evade laborious task in conventional method while keeping the functionality of FEM ABAQUS. A set of parameters, e.g., cross-section geometries, material, slenderness, bolt spacing were determined in this study.

3.1 Introduction

In order to gain a deeper understanding and formulate hypotheses for the structural behavior of semi-closed thin-walled steel columns with polygonal section, a comprehensive parametric study was carried out by means of finite element modelling in ABAQUS.

It is vital to first develop a reliable FE model capable of producing realistic and accurate results, particularly for elastic buckling and non-linear ultimate modes and loads. As the material and geometric non-linear modelling of thin-walled structures is sensitive to modelling inputs and assumptions[**17**], caution should be exercised when defining issues such as the type and size of the element, the material, boundary conditions, imperfections, and solution controls, etc. The model should also be validated before trusting it to generate further data for design purposes. This part follows these principles and describes the essential stages in the development of FE models for columns susceptible to local, distortional, or global buckling, and any possible interactions between these basic buckling modes.

Subsequently, by way of a rigorous and systematic procedure, parametric studies were carried out to produce more than one hundred numerical models. Simulations of the proposed columns subjected to axial compressive load with elastic buckling and non-linear analysis, were performed. Data obtained from the parametric studies mainly included the elastic buckling loads and ultimate strengths of the FE models, as well as load-displacement relations.

The finite element modeling was an integrated process, performed in parallel with parametric study. Both processes were created through automation using MATLAB and Python script. First, the profiles geometrical database of the sections was generated by using MATLAB code. This database was then exported to Python via pickle file. Then the automation was performed in Python environment that will be fed to ABAQUS. The automation process in Python was carried out through back and forth modeling process between ABAQUS/CAE and Python to build the final working models and eventually used as an input file for ABAQUS. This input file of all models was then run

simultaneously in batch mode of ABAQUS platform. Therefore, it can be said that the FE modeling were carried out in fully automation method.

ABAQUS/standard v6.14was used as solvers for the FE simulations, while codes using MATLAB R2015a and Python 2.7.3 were developed to perform the pre-processing (including generating input files) and post-processing tasks. Most simulations were performed on the Cluster with parallel servers provided by LTU's computer lab. The large number of simulations included in the parametric studies would have been impossible without these high performance computing facilities, although the systematic procedure developed to conduct the parametric studies also enabled the successful execution of the large number of analyses.

The proposed procedure will be explained in the following section.

3.2 Automation of Parametric Studies

An FE analysis generally requires three steps: (i) pre-processing to build up the FE model and generate the input file, (ii) job-running by submitting the input file, and (iii) post-processing to extract the results. This study attempted to maximise the level of automation involved in these steps in order to increase the efficiency and capacity of the parametric studies. This automation was mainly achieved using scripts written in the Python language integrated into ABAQUS.

As mentioned above, the job-running step was computationally intensive and was performed in parallel-computing cluster and servers. However, the pre-processing and post-processing steps could be particularly input/output (I/O) intensive yet computationally less intensive, so most operations in these two steps were performed in the local computer.

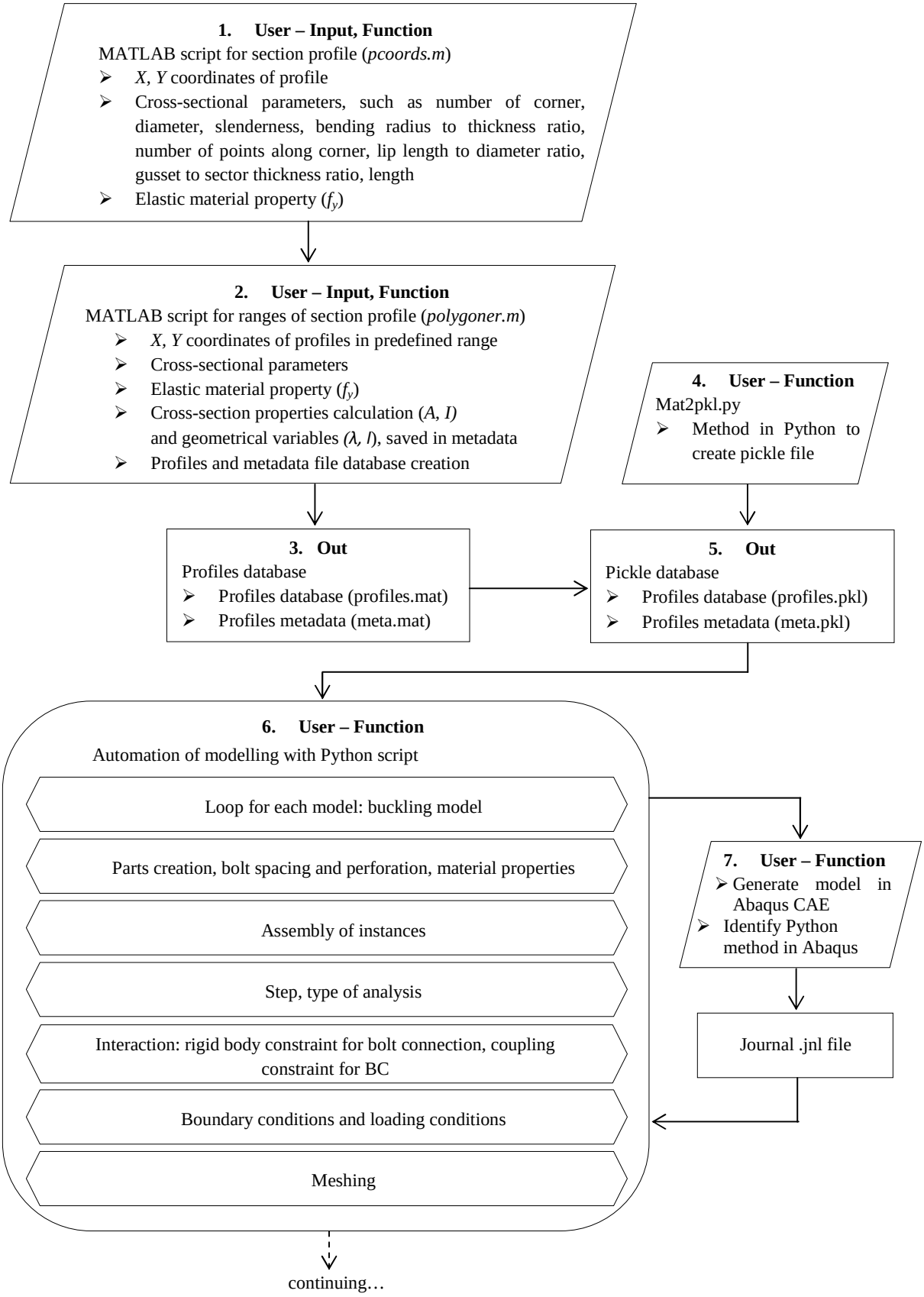
The following flowchart in Figure 37 describes the general procedure used in this research to carry out FE parametric studies for thin-walled steel columns. Efforts were made to increase the level of automation while retaining the accuracy and reliability of the analysis at three key stages, i.e. pre-processing, job running, and postprocessing.

3.2.1 Pre-processing and generation of FE input files

The pre-processing stage in a commercial FE program is normally carried out in a Graphical User Interface (GUI) which requires a significant amount of computing time, which means that this conventional approach is not feasible for a large number of simulations. For this reason an input file generator coded in Python was used where the profiles of section coded in and imported from MATLAB.

In order to generate an input file for a number of particular types of cross-section, the pre-processing process included the following key steps:

(i) Cross-section profiling. A number of cross-sections needed to be modeled for each type of polygonal shape and cross-sectional criteria. The profiles were selected to cover the practical range of applicability and cross-section slenderness values. In the present parametric studies, profiles of all types of cross-section were generated in MATLAB code, bringing out profiles database. This profile database was then converted into a file format i.e. pickle that is compatible for exporting to Python


1. User – Input, Function
MATLAB script for section profile (pcoords.m)
X, Y coordinates of profile
Cross-sectional parameters, such as number of corner, diameter, slenderness, bending radius to thickness ratio, number of points along corner, lip length to diameter ratio, gusset to sector thickness ratio, length
Elastic material property (fy)
2. User – Input, Function
MATLAB script for ranges of section profile (polygoner.m)
X, Y coordinates of profiles in predefined range
Cross-sectional parameters
Elastic material property (fy)
Cross-section properties calculation (A, I) and geometrical variables (λ, l), saved in metadata
Profiles and metadata file database creation
4. User – Function
Mat2pkl.py
Method in Python to create pickle file
3. Out
Profiles database
Profiles database (profiles.mat)
Profiles metadata (meta.mat)
5. Out
Pickle database
Profiles database (profiles.pkl)
Profiles metadata (meta.pkl)
6. User – Function
Automation of modelling with Python script
Loop for each model: buckling model
Parts creation, bolt spacing and perforation, material properties
Assembly of instances
Step, type of analysis
Interaction: rigid body constraint for bolt connection, coupling constraint for BC
Boundary conditions and loading conditions
Meshing
7. User – Function
Generate model in Abaqus CAE
Identify Python method in Abaqus
Journal .jnl file
continuing…

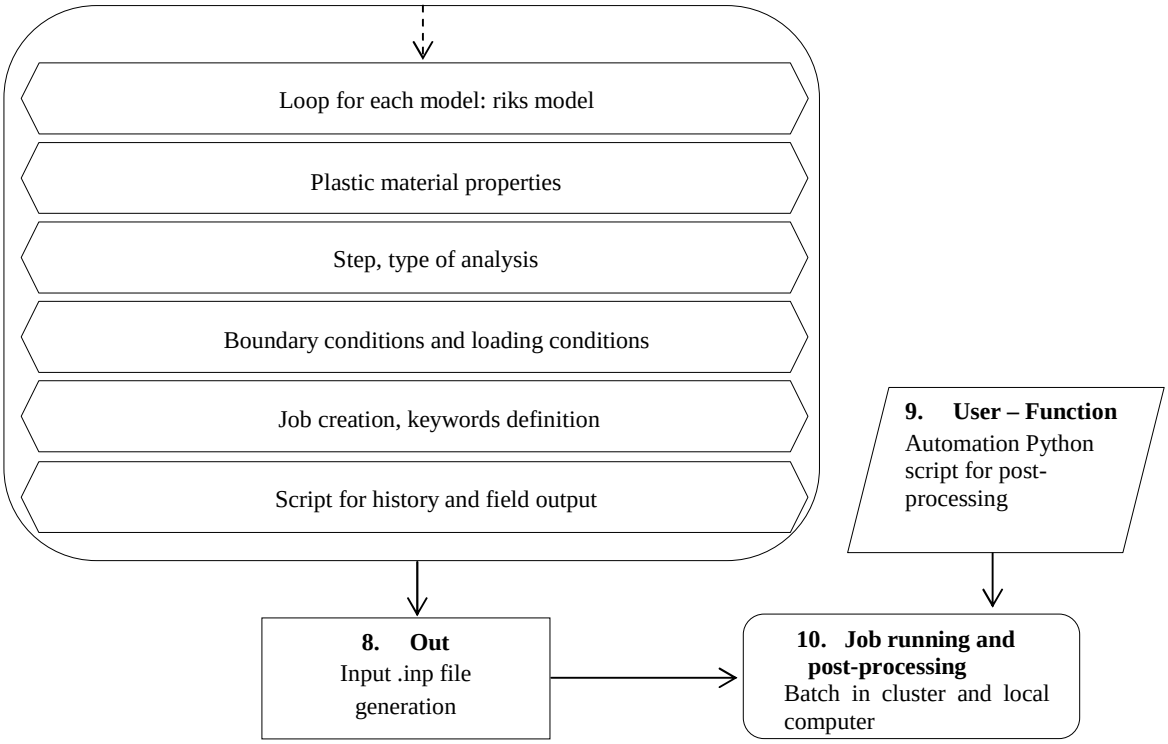


Figure 37.General workflow for FE modeling and parametric studies

(ii) Prepare all parameters for the parametric studies. Set up all requested parameters and initial data. The requested parameters included the cross-sectional characteristics, i.e., polygonal type, diameter, slenderness, bending radius to thickness ratio, number of points along the bend, extension lip length to diameter ratio, thickness of the gusset plate to thickness of sector ratio; material; member length, etc., while the initial data included the cross-section geometrical information, non-dimensional slenderness and all other settings for the input data. Initial data were created in MATLAB together with profile database. This meta database was then converted into a file format i.e. pickle that is compatible for exporting to Python

(iii) Importing profiles database and profiles meta database from MATLAB into Python via pickle file Create automation in Python as input file for ABAQUS. As mentioned previously, a parametric study input file is an inventory that includes the definitions of all the models and jobs for both the elastic buckling and collapse analyses. The calculation of bolt spacing was also performed for each column such that the spacing effect on distortional buckling of the model will be investigated. After that, input file for all models was ready for job running in the cluster. It is worth mentioning that on the script the workflow and explanation, in the form of comments, was included for all relevant coding paragraphs for human readable purpose.

3.2.2 Job running

The optimal way to run simulations depends largely on the resources available. The general availability of CPU, memory, disk storage capacity, will influence the way the simulations were run. Therefore, the job running task used high performance computing facilities provided by LTU. The following points will introduce the general steps involved in the simulation process:

(i) Sub-divide the input file into groups: elastic buckling analyses and collapse analyses

(ii) Sub-divide the input files into different groups according to CPU number required for each job. In this study, the size of the file closely related to the number of partition of parts, the number of elements and hence the DOFs in the model. Number of partitioning became the most significant factor for the file size. Each group served as the basic submission batches that contained several hundred input files, depending on their sizes

(iii) Create batch files. Batch files for all computing facilities were generated simultaneously

(iv) Submit batch files. The number of batches that could be submitted concurrently depended on the capacity of the specific facility with the CPUs, memory, hard disk and running time allowed for each batch may also affecting. Commands were used for submission to a facility operating on Linux.

3.2.3 Post-processing and generation of data

When simulations in each batch of submissions were completed the output result files were downloaded back to the peripheral storage devices of a local PC. Once all the result files for all analysis type of a particular cross-section were collected, the post-processing stage commenced. Its main purpose in this study was to obtain the data set for parametric studies, such as first elastic buckling load and mode, as well as the failure mode and ultimate strength for each simulation. The following paragraph will describe the general post-processing steps of analyses:

(i) Generate a Windows batch file to copy all the result files into one directory

(ii) Extract results. The results for each buckling and collapse analysis were fetched from the ABAQUS.odb files and saved in a text file by using Python script

(iii) Reading each result data to check whether the analysis successfully performed; the job name of any analysis that did not complete was recorded in a text file

(iv) For those jobs not successfully done, find out the reason behind it. Modify the original input file and generate new input file accordingly. Then, loop back the process to step (iii).

Once the result data of all elastic buckling and collapse analyses, it becomes raw data for further analysis. Therefore, these data was turned into final result sheet for columns made from a particular cross-section, generated and classified for each intended parameter. The relevant results were then used for parametric studies presented in Chapter 4.

The ABAQUS scripting is a Python-based application programming interface (API) to ABAQUS. ABAQUS version 6 makes extensive use of Python, a powerful, widely used scripting language to automate repetitive tasks. The diagram below shows the workflow of scripting for ABAQUS.

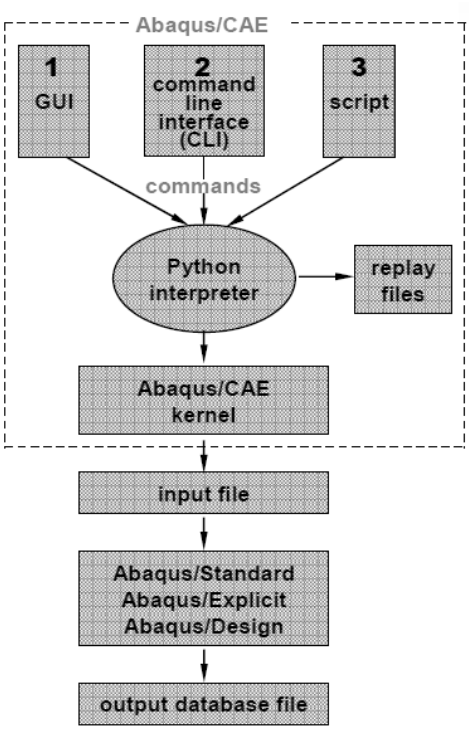


Figure 38. Abaqus Scripting workflow

3.3 Problem Statement

The studied members is chord segment of lattice tower with hexagonal base shape as the common shape for lattice wind turbine tower, hence it forms 120˚ of angle between the horizontal beams. The chord column has three shapes according to number of side or bend, namely 6-sided (hexagon), 9-sided (nonagon), and 12-sided (dodecagon), as shown in Figure 41.

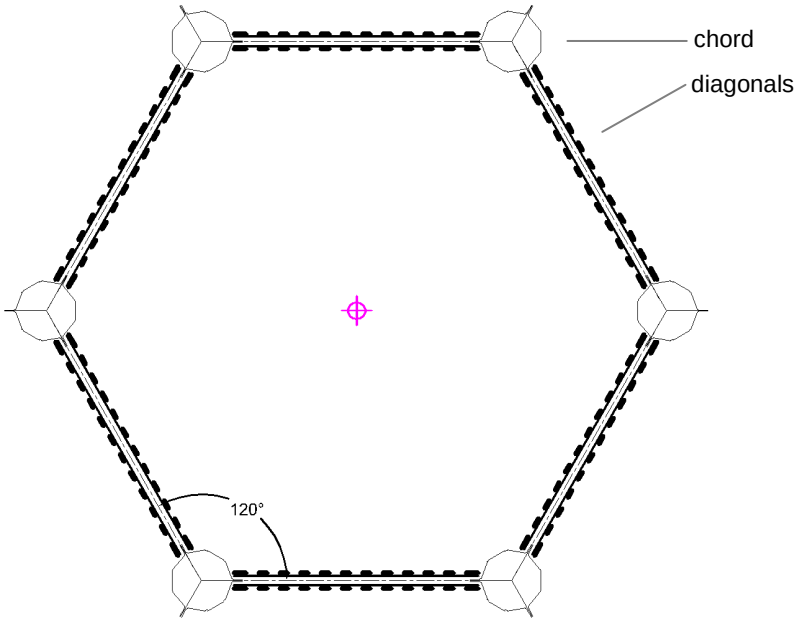


Figure 39.Base of the studied tower

The chord is a built-up member composed of folded plates and gusset plates. The close-up picture of the member is shown in Figure 40. Assembly of the member is performed by bolting them along the lips of the folded plate at a specified spacing. As for the connection between chord and diagonals, gusset plates coming out from the core of the chord provide the joint.

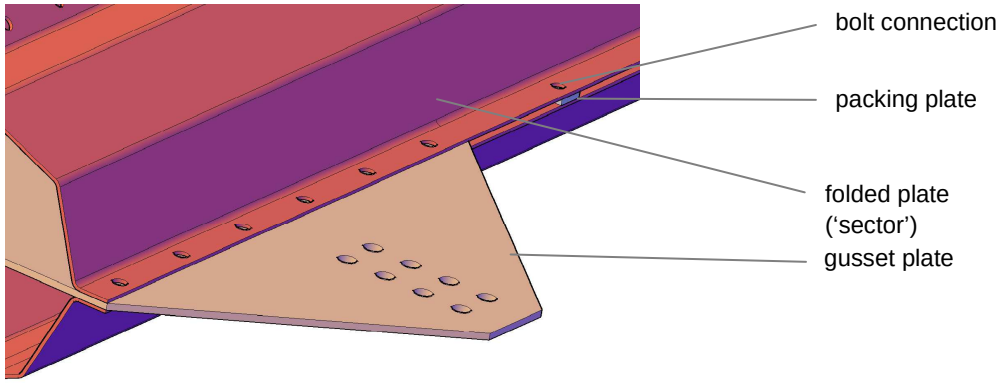
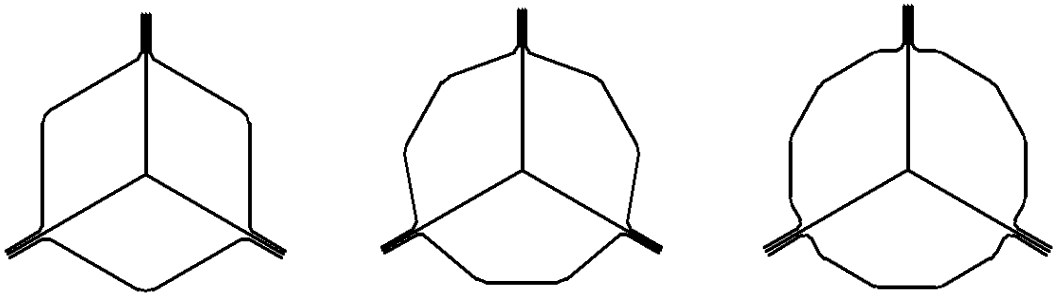


Figure 40. Closed-up of the member



(a) 6-sided hexagon (b) 9-sided nonagon (c) 12-sided dodecagon

Figure 41. Type of polygonal cross-section for the studied chord

As can be seen in Figure 42, forces acting on diagonals may create bending moment at the connection with the chord, and consequently utilizes rotational capacity of the joint. This aspect will be further investigated in Chapter 4.

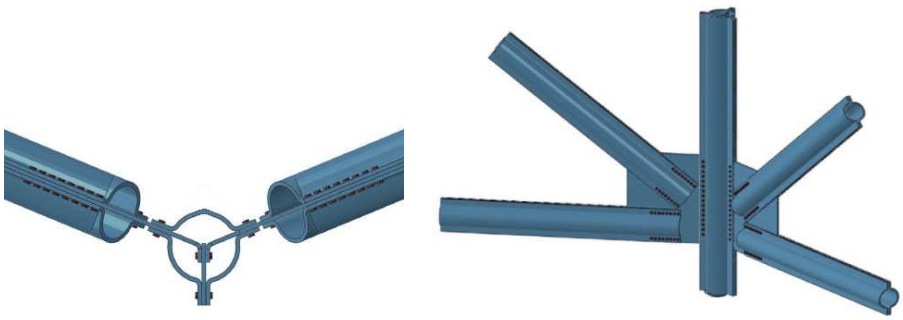


Figure 42.Illustration of chord-to-diagonals bolted connection

3.4 Development of FE models

3.4.1 Units

Table 4. Units used for FE analysis

Length	Force	Stress	Energy	Density	Young's Modulus	Gravity
mm	N	MPa	N-mm	7,83e-09	2,07e+05	9,806e+03

3.4.2 Geometry and Material Properties

3.4.2.1 Geometry

The proposed members are built-up polygonal cross-sections. Each polygonal section is composed of three folded plates which forming 120˚ angle of gusset plate. This configuration is intended as a chord member of lattice tower with hexagonal form of base, as shown in Figure 39. The connection between chord and diagonals are possibly made through the gusset plate extending on the two sides of the cross-section. The folded plates in assembly are separated by gusset plates at the connection segment; by packing plates at the bolt connection points along the length; and by gaps at the remaining regions.

The geometry of section profiles was generated through automation using MATLAB code. The workflow of the software built by MATLAB will be described in the following sections.

- ***File name: pcoords.m***

This code is basically the engine for generating *x, y* data of the profile. First of all, function script for profile of one folded plate (henceforth called sector) was made, with input variables: number of corner (*n*), diameter of cross-section (*d*), profile slenderness (*slend*), yield strength (f_y), bending radius to thickness ratio (*rcoef*), number of points along the bend (*nbend*), lip length to diameter ratio (*l_ratio*), gusset to sector thickness ratio (*t_ratio*).

```
function [x_out, y_out, t, tg, l_lip] = pcoords(n, d, slend, fy, rcoef,
nbend,l_ratio, t_ratio)
```

The output variables are:

x_out, *y_out*	is *x, y* coordinate of a profile according to the given input variables
t	is thickness of profile according to cross-sectional slenderness ($\varepsilon^2.d/t$)
tg	is thickness of gusset plate according to *t_ratio* (t_g/t) variable
l_lip	is length of lip extension according to *l_ratio* (*l_lip/d*) variable

The above output variables are used to generate profile of sector plate, as can be seen in Figure 43-45 below.

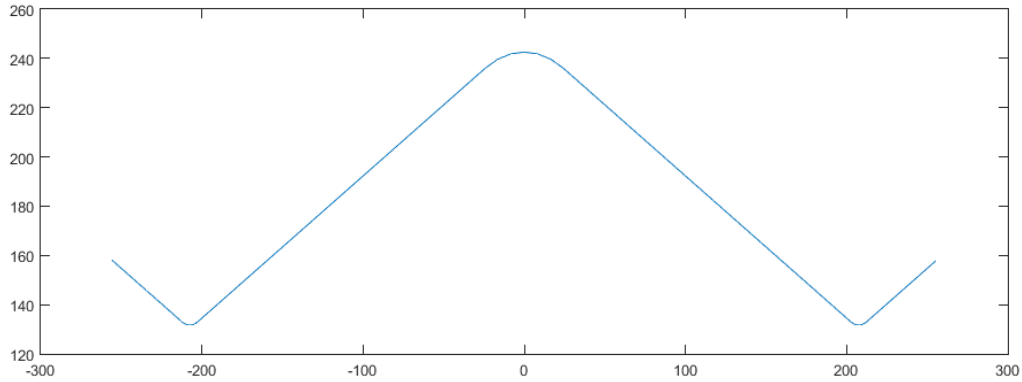


Figure 43.Figure of profile generated by MATLAB (n=6, d=500, slend=90, fy=355, rcoef=6, nbend=5)

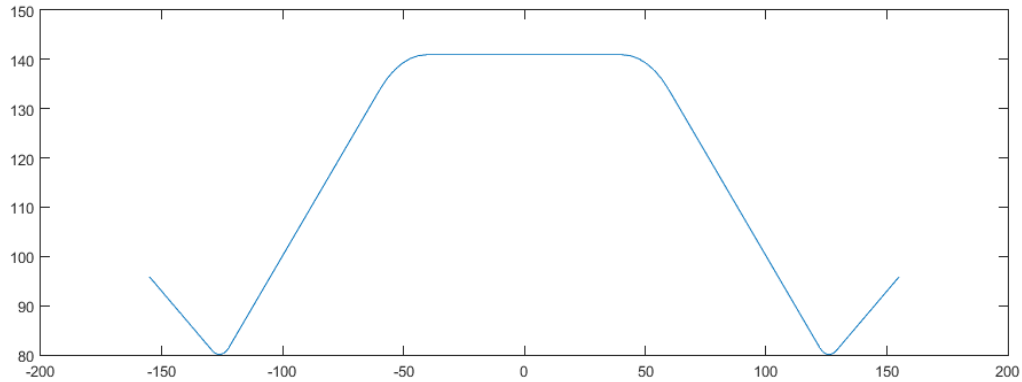


Figure 44.(n=9, d=300, slend=90, fy=355, rcoef=6, nbend=5)

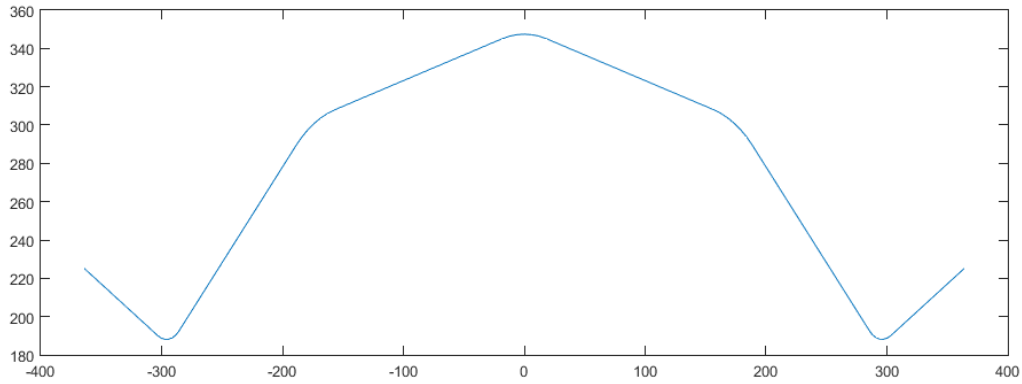


Figure 45. (n=12, d=700, slend=90, fy=355, rcoef=6, nbend=5)

The function script contains codes of geometrical calculation to generate [*x*,*y*] coordinates of the profile based on the predefined geometrical parameters. The step-by-step explanation of the codes is presented here.

>> Line 1-2

Function script, contains output variables and input variables of the function.

>> Line 3-36

The script contains recommended input argument. Descriptions of each input argument are also given.

>>Line 40-45

Characteristics of profile were defined here. Thickness of profile is calculated according to cross-sectional slenderness (*slend*) as a function of diameter and epsilon. The slenderness range was chosen so as the cross section class to be in the limit between Class 3 and 4, based on EC 3-1-1.

$$70 < \frac{d}{\varepsilon^2\, t} < 150$$

The thin-walled profile usually made of plates which fall in Class 4 cross-section. However, in order to limit the complication of effective cross section in calculation, profile in between these cross-section classes is chosen. In the parametric study, only two values of profile slenderness were analysed, i.e. 90 as the lower bound and 110 as the upper bound. Thickness calculation is rounded to get integer value.

```
% Calculated characteristics
R = d/2;
epsilon = sqrt(fy/235);
t = round(epsilon^2 * d / slend);
tg = round(t_ratio*t);
l_lip = l_ratio*d;
```

>>Line 46-142

The script contains calculations for the *x*,*y* coordinate to produce profile of a sector. The step can be summarized, as follows:

① Calculate center of polygon (*0, 0*) →② calculate polygon's corner (x_i, y_i) →③ calculate center of bending arc (x_c, y_c) →④ calculate *x, y* coordinate of the arch's points (x_{arc}, y_{arc}) →⑤ loop →

⑥ calculate lip's center of bending arc (x_{cs}, y_{cs}) →⑦ calculate x, y coordinate of the lip's arch's points (x_{sarc}, y_{sarc}) →⑧ loop →⑨ calculate point of the lip (x_{start}, y_{start}) and (x_{finish}, y_{finish})

```
% Angle corresponding to one edge of the polygon
theta = 2*pi/n;
% Angles of radii (measured from x-axis)
phi=5*pi/6:-theta:pi/6;
% xy coords of the polygon's corners
x = R*cos(phi);
y = R*sin(phi);
```

Theta (θ) is the relative angle between lines, while phi (φ) is the absolute angle of points with respect to *x*-axis; *n* is number of corners. From the phi angle the coordinate of polygon's corners can be calculated. Detail of geometry calculation can be seen in Figure 46-48.

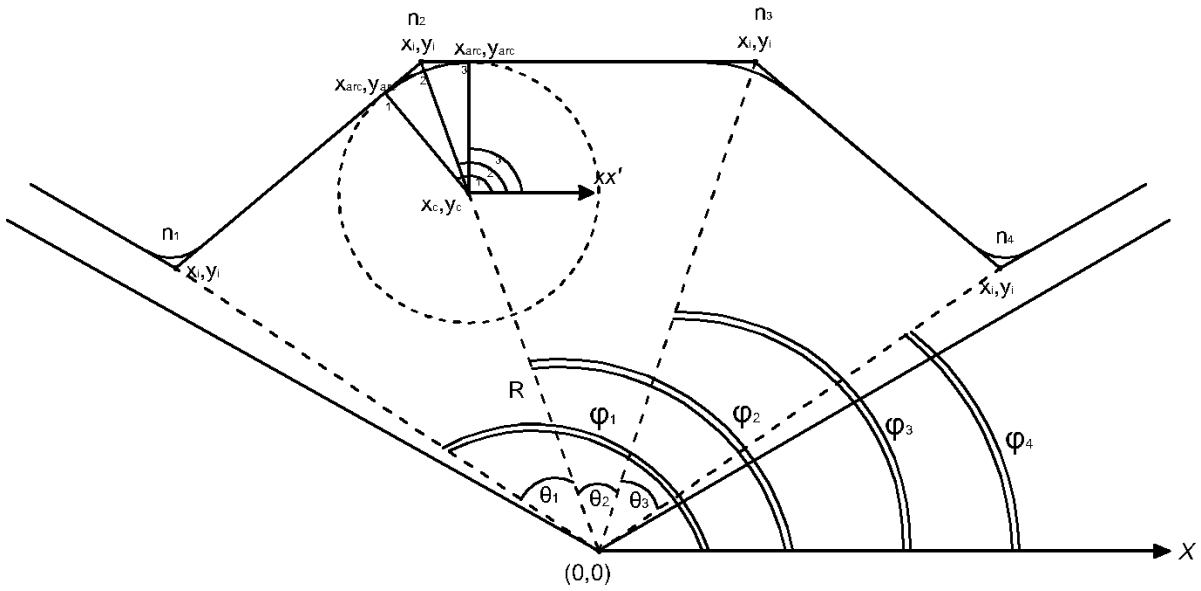


Figure 46. Geometry calculation

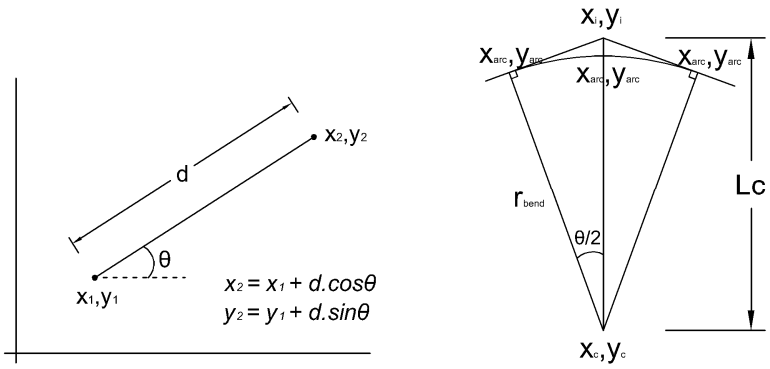


Figure 47. Inset of geometry calculation

The center of bending arcs (x_c, y_c) then calculated with refer to the obtained x, y values, by using Pythagorean theorem. This center point become a reference for calculation of x, y coordinates along the bent (x_{arc}, y_{arc}).

```
% Bending radius
rbend = rcoef*t;
% Distance between bending centre and corner
lc = rbend/cos(theta/2);
% Centers of bending arcs
xc  = (x(2:end-1) - lc*cos(phi(2:end-1)));
yc  = (y(2:end-1) - lc*sin(phi(2:end-1)));
% Angles of the edges' midlines (measured from x-axis)
phi_mids = phi(1:end-1) - theta/2 ;
```

Loop function was used to generate these arc's points as many as *nbend* input argument. Angle of the midline (φ_m) was used as reference angle for calculating each arch's point.

```
% xy coords of the arc's points
for i = 1:n/3 -1;
for j = 1:nbend+1;
        xarc(i, j) = xc(i) + rbend*cos(phi_mids(i)-(j-1)*(theta/nbend));
        yarc(i, j) = yc(i) + rbend*sin(phi_mids(i)-(j-1)*(theta/nbend));
end;
end;
```

The calculation continued to the lip extension part, with the same workflow to the main part. Corner of polygon (x_1, y_1) with refer to corner point at gusset plate location (x, y) was first calculated according to angle of midline (φ_m). From this, the center of bend (x_{cs}, y_{cs}) can be calculated using the angle of adjacent sector (v). After that, the points along bend (x_{sarc}, y_{sarc}) were calculated based on center of bending and angle of the vector with respect to x-axis.

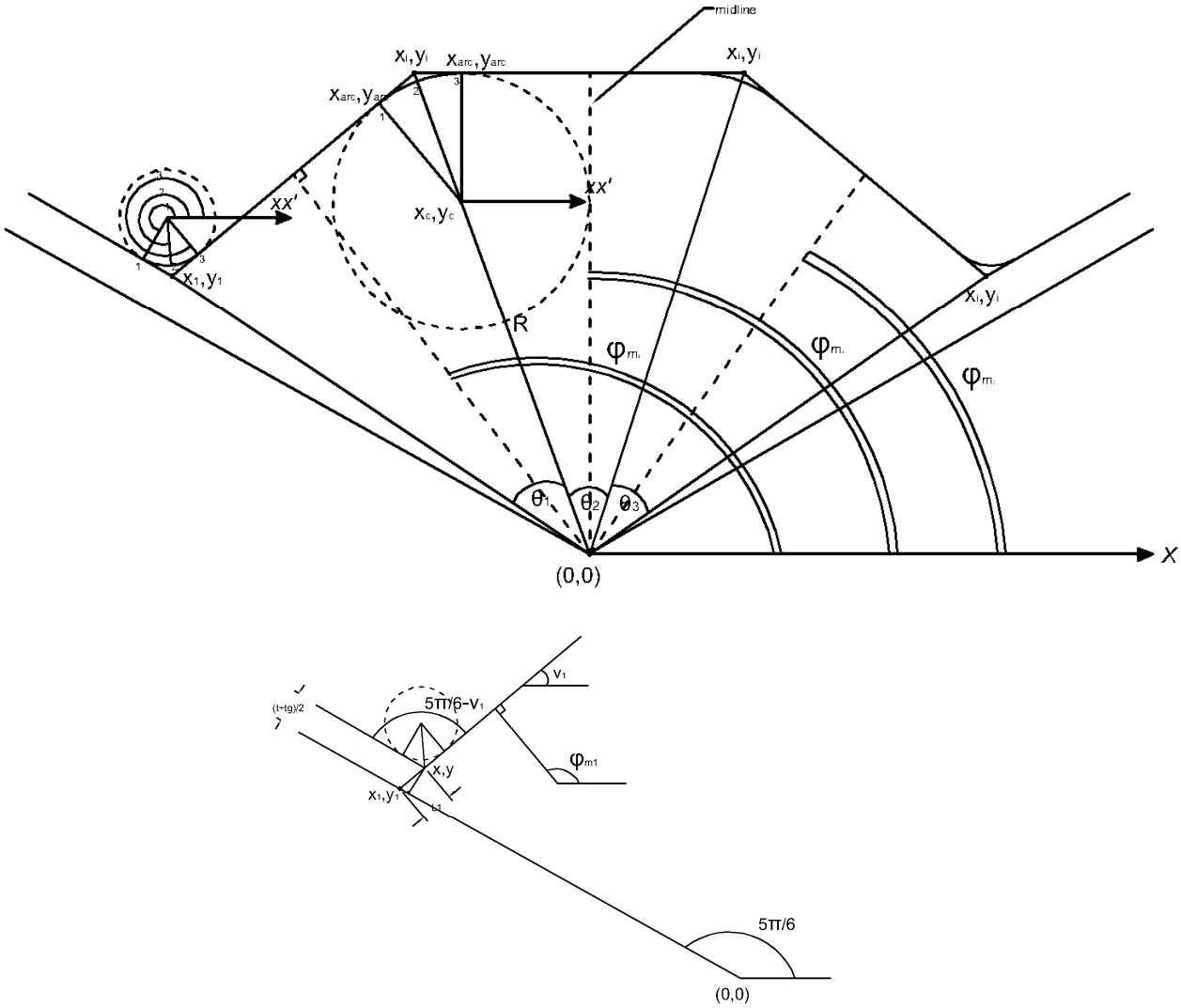


Figure 48. Geometry calculation for extension lip

```
%% Start-end extensions
% Bending radius
rs = rbend/5;
% First bend
v1 = phi_mids(1)-pi/2;
v2 = (phi(1)+phi_mids(1)-pi/2)/2;
l1 = (t+tg)/(2*cos(phi(1)-phi_mids(1)));
l2 = rs/sin(v2-phi_mids(1)+pi/2);
x1 = x(1)+l1*cos(v1);
y1 = y(1)+l1*sin(v1);
% First bend centre coords
xcs(1) = x1+l2*cos(v2);
ycs(1) = y1+l2*sin(v2);
```

The bend center coordinate for end bend can be calculated in the similar way. Loop function was used to generate these arc's points as many as *nbend* input argument.

```
% First and last bend arc points coords
for j = 1:nbend+1;
    xsarc(1, j) = xcs(1) + rs*cos(4*pi/3+(j-1)*((phi_mids(1)-
pi/3)/nbend));
    ysarc(1, j) = ycs(1) + rs*sin(4*pi/3+(j-1)*((phi_mids(1)-
pi/3)/nbend));
    xsarc(2, j) = xcs(2) + rs*cos(phi_mids(end)+pi+(j-1)*((phi(end)+pi/2-
phi_mids(end))/nbend));
    ysarc(2, j) = ycs(2) + rs*sin(phi_mids(end)+pi+(j-1)*((phi(end)+pi/2-
phi_mids(end))/nbend));
end;
```

The next part of calculation is to define the *x, y* coordinate for the lip ends. These two points will be the start and end point of profile coordinates, respectively. Pythagorean theorem as per Figure 48 was used to calculate this point.

```
%% Points of the lips
% First lip
xstart = [xsarc(1, 1) + l_lip*cos(phi(1)), xsarc(1, 1)
+l_lip*cos(phi(1))/2];
ystart = [ysarc(1, 1) + l_lip*sin(phi(1)), ysarc(1, 1)
+l_lip*sin(phi(1))/2];
```

The last point of lip can be calculated accordingly. Finally, the last step is to collect all points that were calculated into a sorted array. Coordinates along the flat plate (*xstart, xend; ystart, yend*) and corners (*xsarc, xarc; ysarc, yarc*) of the profile were generated, as follows.

```
x_out = [xstart, xsarc(1, :), xarc(:)', xsarc(2, :), xend];
y_out = [ystart, ysarc(1, :), yarc(:)', ysarc(2, :), yend];
```

The complete *pcoords.m* script for generation of *x,y* coordinate can be found in Annex A.1 line 1 – 132.

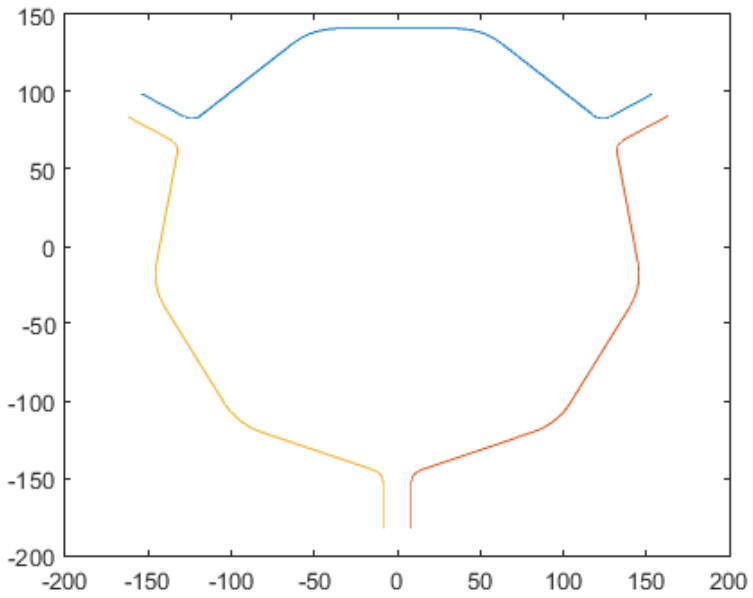


Figure 49. Figure of assembled profiles generated by MATLAB (n=9, d=300, slend=90, fy=355, rcoef=6, nbend=5)

Figure 48 and 49 shows that the calculation takes into account the actual gap between profiles which comes from half the thickness of sector profiles and gusset plate thickness. This arrangement has been made to make sure that there is neither overlap nor gap when all the parts are assembled. The calculation also related to the thickness extrusion method used in ABAQUS, which in this case uses middle plane extrusion.

Thickness of gusset plate was locked to gusset-to-sector thickness ratio (*t_ratio*), while thickness of sector was locked to cross-sectional slenderness (*slend*).

The above *pcoords.m* script was then further developed and used to create another script, named *polygoner.m*, in order to generate profile coordinates for ranges of parameter values. The script was also intended to create metadata file, for the purpose of storing geometry variable database needed for modelling automation in Python.

- ***File name: polygoner.m***

This code is the core software for generating several profile data and variable database in range of input parameters.

>> Line 1-2

Function script was created with input variables: range of *n*, *d*, *slend*, and *lambda*; single value of *fy*, *rcoef*, *nbend*, *l_ratio*,and *t_ratio*.

```
function [profiles, meta] = polygoner(nrange, drange, slendrange, fy, rcoef, nbend,
l_ratio, t_ratio)
```

The output variables are:

profiles	is a cell containing arrays of *x*, *y* coordinates of profiles
meta	is a cell containing arrays of variable values, e.g., diameter (*d*), thickness (*t*), gusset thickness (*tg*), yield strength (*fy*) cross section properties (*A*, *Iyy*, *Izz*), and length (*len*)

>> Line 3-26

The script contains recommended input argument of range. Descriptions of each input argument are also given.

>> Line 28-33

Briefly, the script will generate two databases: profiles and meta, which contain arrays with the same dimension.

```
% Initialise a cell array to host the profiles' data
profiles = cell(length(nrange), length(drange), length(slendrange));
meta = cell(length(nrange), length(drange), length(slendrange));length(lambda));
```

The range and single value of input arguments were determined as per Table 6. There is one new variable in the function script, i.e. *lambda*. This variable is non-dimensional slenderness, which determines the critical buckling length of a member.

$$\bar{\lambda} = \sqrt{\frac{A.f_y}{N_{cr}}} \quad ; \; N_{cr} = \frac{\pi^2 E I_y}{{l_{cr}}^2}$$

$$\bar{\lambda} = \left(\frac{A.f_y.l^2}{\pi^2 E I_y}\right)^{0.5}$$

Variable *lambda* will be used to calculate length of the profiles. It will take the information stored in meta database, e.g. cross-section properties.

>> Line 35-44

The process was done by running loop through the values within the given ranges. The `for…end` command was used to make loop of profiles, with *i*, *j*, and *k* as range of numbers for *n*, *d*, and *slend* parameter, respectively. Firstly, it calls *pcoords* function to get data for profile and then collect *x*, *y* coordinates of profiles into profiles database.

```
% Loop through the values within the given ranges
for i = 1:length(nrange);
for j = 1:length(drange);
for k = 1:length(slendrange);
            % Call pcoords to get data for a profile
            [x, y, t, tg] = pcoords(nrange(i), drange(j), slendrange(k), fy,
            rcoef, nbend, l_ratio, t_ratio);
            % Collect the xy values in a database
            profiles{i, j, k} = [x; y];
```

>> Line 46-110

Secondly, calculation for profiles meta database was performed. Calculation of cross section properties were made by using *cutwp_prop2* function module which available in CUFSM software package. In order to use this function, codes to create nodes and element arrays for the profiles as an input to *cutwp_prop2* function should be made. Process of constructing nodes and elements can be seen in the complete script line 52 – 84.

```
% Return cs properties using cutwp
[A, ~, ~, Iyy, Izz, Iyz] = cutwp_prop2(node, elem);
% Current profile area and moment of inertia
I = min(Iyy, Izz);

% Loop through the different member slendernesses. The 'meta'
% array has one more dimension (4D)
for l = 1:length(lambda);
    % Current profile length
    len = lambda*pi*sqrt(E*I/(A*fy));
    % Store the metadata in a cell array
    meta{i, j, k, l} = [drange(j); t; tg; fy; A; Iyy; Izz; Iyz];
    len(l)];
end
```

The cross sectional properties (*A*, *Iyy*, *Izz*, and *Iyz*) are for the entire assembled cross-section. As can be seen in the script, the cross sectional properties are needed for slenderness (λ) calculation, which will be used for determining the length of member and lock it based on certain slenderness value defined in Python automation. These variables (*drange(j); t; tg; fy; A; Iyy; Izz; Iyz*) were stored in profiles meta database.

At the end of the script, a command was written to create .mat files for each database function. Later, these files will be converted into pickle file using *mat2pkl* Python method, and used in Python automation.

```
% Save the profile database and metadata to the current directory as .mat
save('profiles.mat', 'profiles');
save('meta.mat', 'meta');
```

In this script, calculations for cross-section class and the corresponding effective area were also performed. This data are required for the analytical calculation based on Eurocodes. It can be found in line 81 – 133.

The complete script for generation of several profiles and meta database in predefined range can be found in Annex A.2 line 1 – 155.

Ranges of values for each parameter used in this parametric study are tabulated in the following table. In order to complete the FE modelling and analyses optimally in the predefined time, the number of input variables used in this parametric study were reduced, so that only some input variables of the parameter were taken from the database.

Table 5. Ranges of parameter values

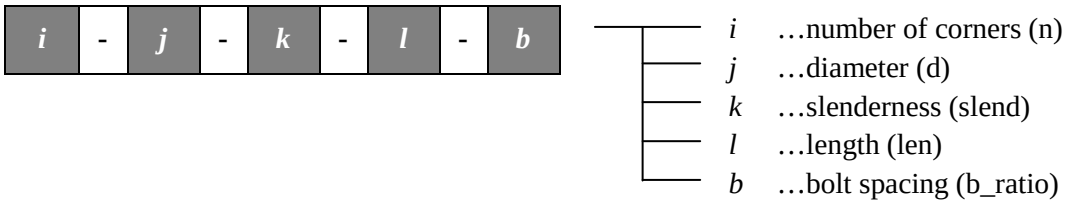
Parameter	Description	Function	Value(s)	Number of model for each profile	Total number of models acc.
n	number of corners	[6, 9, 12]	6	3	18
d	diameter of chord	[300:200:900]	500, 700, 900	3	
slend	profile slenderness	$d/(\varepsilon^2.t)$ linspace(70,150,10)	90, 110	2	
fy	yield strength	355	355	1	
rcoef	bending radius to thickness ratio	6	6	1	
nbend	number of points along the bend	5	5	1	
l_ratio	lip length to diameter ratio	0.1	0.1	1	
t_ratio	gusset to sector thickness ratio	1.2	1.2	1	
len	length of chord	$\bar{\lambda} = \left(\frac{A.f_y.l^2}{\pi^2 E I_y}\right)^{0.5}$	0.65, 1.0, 1.25	3	
b_ratio	bolt density	s/d	3.0, 4.0, 5.0	3	54
N	Axial force	-	N	1	54
NM	Axial force-bending moment	-	0.05M, 0.1M, 0.15M	3	216

From Table 5 above it can be seen that there are one other parameter besides the ones defined in MATLAB function script above. This parameter is bolt density (*b_ratio*). This parameter, together with length parameter, are important in the buckling behavior of the member since they affect the buckling half-wave length of the member which in turn becomes decisive parameters for the buckling and interaction buckling failure mode, as described in Chapter 2.

The length of chord is calculated by keeping the non-dimensional slenderness $\bar{\lambda}$ to certain values, which are 0.65, 1.0, and 1.25. These values were chosen to make sure the member will not either be too slender or too stocky. The calculation require cross-section properties: area of closed cross-section, *A*, and moment of inertia, *I* and this was carried out in MATLAB script and stored in meta file which then used in the Python script.

The bolt density as function of spacing-to-diameter ratio was fully calculated in Python, together with the creation of bolt holes. The s/d ratio was determined to observe the influence of bolt density to distortional and local buckling occurrences, which in this study was taken 3.0, 4.0, and 5.0. This part will be further explained in the section of perforation modeling.

Five parameters which have varied values: *n*, *d*, *slend*, *len*, and *b* determine the naming of numerical model. In order to make easy identification, it was decided to name the models with numbers, as follows.



The variable number for naming and the corresponding value is shown in the table below.

	i			**j**				**k**		**l**			**b**		
var. ID	1	2	3	1	2	3	4	3	5	1	2	3	1	2	3
values	6	9	12	300	500	700	900	90	110	0.65	1.0	1.25	3	4	5

- ***Conversion from MATLAB to pickle file***

Profiles database calculated in MATLAB were then exported to Python script, as the scripting language used in ABAQUS. The automation then carried out in Python directly. The database file exporting from MATLAB to Python was performed via pickle file.

The creation of pickle file was done through *Mat2pkl.py*module. It is essentially a small module for python. The steps can be summarized below:

1. Save the database file as .mat files
2. Run a python 2.7.3 command prompt
3. Make sure that the .mat database is in the python working directory
4. Import the method mat2pkl from the mat2pkl file
5. Run the method for a given filename. The filename should be given without the extension
6. Example code:

```
from mat2pkl import mat2pkl
mat2pkl("filename")
```

Script of *Mat2pkl.py* can be written as follows:

```
# Python method converting a 3D cell array from a matlab file (.mat) to
anequivalent pickled object containing 3D nested lists.
# This script has to be executed by an generic Python of the same
versionas the one in Abaqus (i.e 2.7.3). Requires pickle and scipy
# The filename is given without the extension

# Module imports
import scipy.io as sio
import pickle

def mat2pkl(filename):
# Load the matlab file
    database = sio.loadmat(filename+'.mat')
```

```
# Scipy imports the data of the .mat in a dictionary.
# Get the lists from inside the dictionary
    ppp = database[filename]

# Export with pickle to a .pkl file
    pickle.dump(ppp, open( filename+".pkl", "wb" ))
```

Each model represents a corner chord member of lattice tower and was created as a segment, consist of three connection parts, located at the ends of segment and at the middle, and chord in between the connection parts. The configuration is shown in Figure 50.

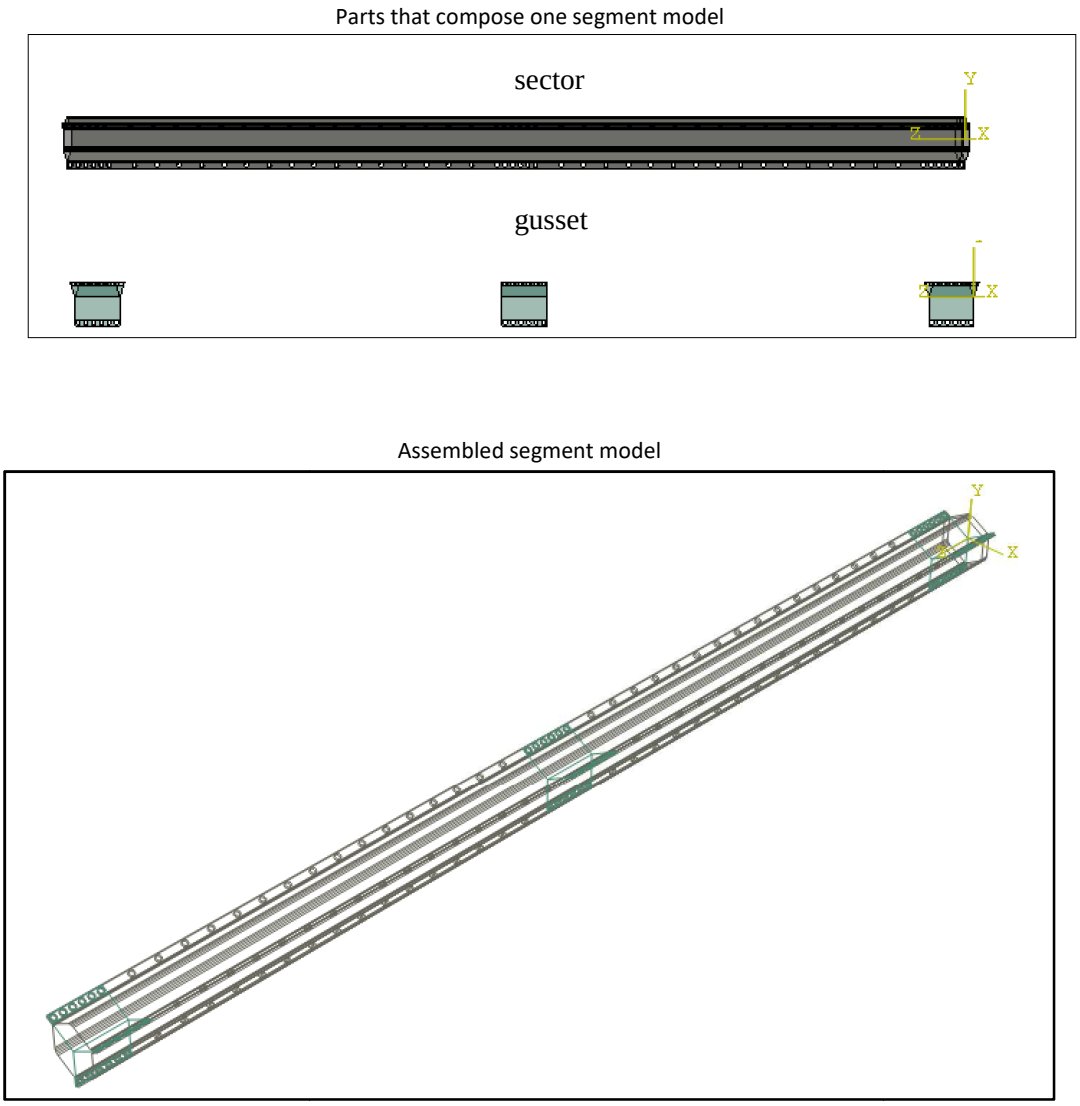


Figure 50. Segment model

3.4.2.2 Material properties

The material property given by EC3 for cold-formed profile and used in the normal production is S355, which properties are summarized in the following table:

- Elastic

Table 6. Steel property used

Type of steel	Grade	F_{yb} [N/mm^2]	F_u [N/mm^2]
Non-alloy structural steel	S 355	355	510

Nominal values are used for the elastic characteristics of steel - the Young's modulus E = 210 000 N/mm^2, and poison's ratio is 0.3.The following recommendations have to be respected:

- F_u/f_{yb} = 1,10
- Elongation at failure = 15%
- $\varepsilon_u = 15\ \varepsilon_y$

- Plastic

The relationship between yield stress (MPa) and plastic strain (%) was defined based on uniaxial coupon test data from Complab LTU. The testing machine measured total strain and reaction value of each specimen from elastic range until failure. It is worth noting that the true stress-strain data was used as input into ABAQUS as a series of data points.
The plasticity table is taking values without the elastic part. In essence, the plastic strain is obtained by subtracting the elastic part from the total strain:

$$\varepsilon_{tot} = \varepsilon_{el} + \varepsilon_{pl}$$

$$\varepsilon_{pl} = \varepsilon_{tot} - \varepsilon_{el} = \varepsilon_{tot} - \frac{\sigma}{E}$$

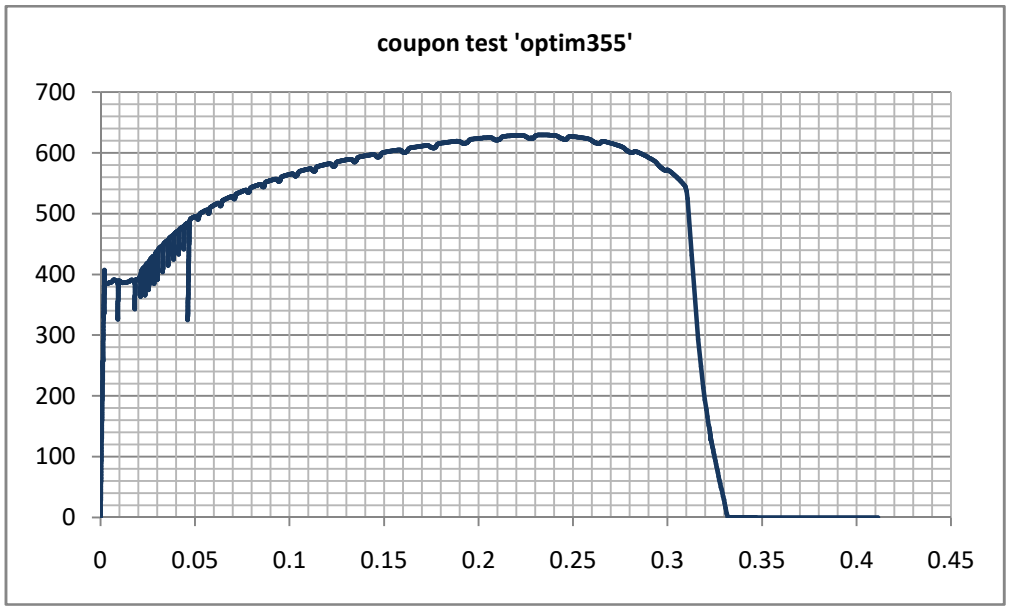


Figure 51. Coupon test stress-strain curve of S355 steel specimen

Step of constructing plasticity table for the use in FEM can be summarized as follows.

1. Take specific stress at plastic zone from true stress-strain curve
2. Take the corresponding total strain with excluding elasticity (use tangent of elastic portion, E, to calculate elastic strain)
3. Add a few points on the plastic part (e.g. in range 0% - 14%) while keeping the last two points has a moderate inclination. This is due to the reason that ABAQUS will continue the curve by extrapolating these two values. A steep increase of this part may cause over-strength in the FEM analysis.

Table 7. Plastic material properties

#	Yield stress	Plastic strain	#	Yield stress	Plastic strain	#	Yield stress	Plastic strain
1	381.1	0	4	418.0	0.0228	7	539.1	0.0765
2	391.2	0.0053	5	444.2	0.0310	8	562.1	0.1009
3	404.8	0.0197	6	499.8	0.0503	9	584.6	0.1221
						10	596.4	0.1394

3.4.3 Python automation

3.4.3.1 Introduction

The ABAQUS Scripting extends Python with several types of objects. The hierarchy and the relationship between these objects is called the ABAQUS Object model.

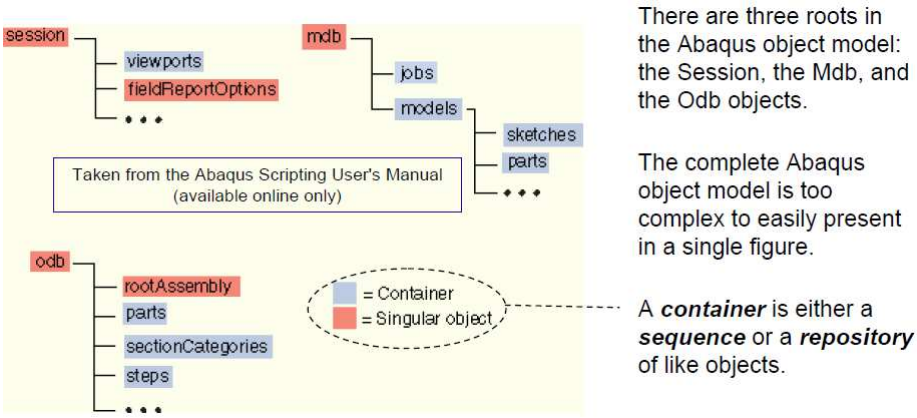


Figure 52. ABAQUS object model

The data encapsulated by an object are called the **members** of the object. The functions that manipulate the data are called **methods**. A method that creates an object is called a **constructor**. **Ownership** defines the access path to the objects. Any Python statement that accesses the Session, Mdb, or Odb is called a **command**.

Commands are used to access objects by stepping through the hierarchy of objects in the object model. All commands mirror the structure of the object model. For example:

```
vp = session.viewports['Viewport: 1']
vp.setValues(origin=(200,0))

cell4 = mdb.models['block'].parts['crankcase'].cells[4]
```

Variable **cell4** | Model database **mdb** | model named **block** | part named **crankcase** | cell with index 4

It can be interpreted that the *cell* with an index of *4* in the *part* named *crankcase* in the *model* named *block* in the model database *mdb* is assigned to the variable *cell4.*

The first script need to be defined is importing Abaqus modules. Each module has functionality attributed to it. Modules are imported using the import statement, the same as any other Python module:

```
from caeModules import *
```

Table 8. Modules and their functionality

Kernel Module	Functionality
part	Creating and modifying parts
material	Creating and modifying materials
section	Creating and modifying sections
assembly	Creating and modifying assemblies
step	Creating and modifying steps
interaction	Creating and modifying interactions
load	Creating and modifying loads and boundary conditions
mesh	Meshing part instances
job	Creating, submitting, and monitoring jobs
visualization	Visualizing results
odbAccess	Accessing the output database file
sketch	Creating and modifying sketches

The script for this part can be seen in Annex A.4 line 1 – 15.

3.4.3.2 Part Module

As mentioned before, the modelling of section members were carried out through fully automation process by using Python script. All parts were modeled using shell element, 3D deformable type with extrusion feature. Instead of creating sketch of profile in CAE environment, the profiles were imported from database file in pickle format (.pkl) stored in working directory.

Workflow of the modeling process in Python for part module can be described as follows.

1. Import database pickle files required for profile definition (profiles database and profiles metadata) which stored in working directory
2. Create the models in a loop. Create models instead of renaming to make more tidy and straight code
3. Create different models for buckling and Riks analysis. First create the buckling models and then copy it into the Riks models. They will be identical in shape but they will differ in the boundary conditions, the loads, the keywords etc. This way done so that they can have individual Jobs for the buckling and the Riks analyses and have individual input files

4. Extrude the sector part based on geometry calculation in profiles metadata file
5. Create gusset part directly in Python based on geometry calculation in profiles metadata file

Each model consists of two parts, 'sector' and 'gusset' part. Sector part represents the folded plate and was profiled in MATLAB script, which then imported in Python. Gusset part represents the gusset plate at connection regions, protruding from the center of assembly cross-section to the chord-diagonals connection.

- ***Sector***

The script for modeling parts in loop, in accordance with the workflow mentioned above is written below. Thicknesses of sector part are locked based on width-to-thickness ratio (*slend*) parameter. The calculation was performed in MATLAB script.

$$\boldsymbol{t = \frac{d}{\varepsilon^2 . slend}}$$

This ratio refers to the cross-section classification regulated in Table 5.2 of EC3-1-1[**9**].
Length of sector parts are locked based on non-dimensional slenderness ($\bar{\lambda}$) parameter. The calculation was performed in Matlab script.

$$\boldsymbol{len = \lambda . \pi \left(\frac{E . I}{A . f_y} \right)^{0.5}}$$

In Python, profiles will be extruded based on this calculation with total length equal to the length of one segment, composed of two chords and three gusset plates.

$$\boldsymbol{l_{tot} = 2 . l_{chord} + 3 . l_{gusset}}$$

One peculiarity in Python is that the indexing starts from 0 and not from 1 as it is usually, so a list of 10 items numerates [0, 1, 2, 3, ..., 8, 9].

```
## 1st Phase: Buckling analysis
for i in range(profiles.shape[0]):
   for j in range(profiles.shape[1]):
       for k in range(profiles.shape[2]):
           for l in range(profiles_meta.shape[3])
                # Variables holding information of the current profile
                current_model=str(i+1)+'-'+str(j+1)+'-'+str(k+1)+'
'+str(l+1)
                current_d = float(profiles_meta[i][j][k][l][0][0])
                current_t = float(profiles_meta[i][j][k][l][1][0])
                current_tg = float(profiles_meta[i][j][k][l][2][0])
                current_fy = float(profiles_meta[i][j][k][l][3][0])
                current_l = float(profiles_meta[i][j][k][l][7][0])
                current_llip = sqrt((profiles[i][j][k][0][0]-
                profiles[i][j][k][0][2])**2+(profiles[i][j][k][1][0]-
                profiles[i][j][k][1][2])**2)
                # Create model ------------------------------------------
                # Create Parts ------------------------------------------
                # Sector
                # -Profile sketch for sector
                # -Sketch sector lines
                # -Extrude sector part
```

with,
i refers to input variable number of corners
j refers to input variable diameter
k refers to input variable cross-section slenderness
l refers to input variable length

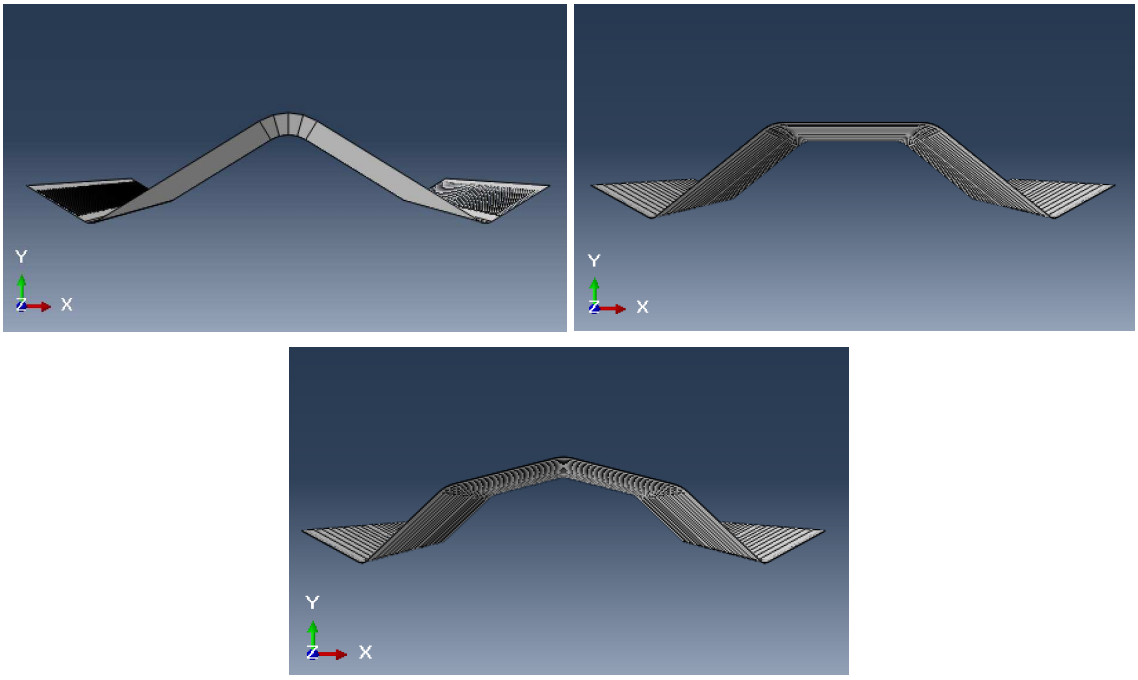


Figure 53. Created sector part (n=6, 9, 12)

The detail script for creating sector part can be seen in Annex A.4 line 51 – 90.

- ***Gusset***

The gusset plate has to be adjusted according to the profile's diameter. Unlike the sector part, profile of gusset part was created directly in Python due to fixed shape of the gusset for all sections. The variable need to be calculated is radial length of gusset from the center point [*0.0, 0.0, 0.0*] to the tip of lips. This calculation is a bit tricky since the radius of assembly section, apart from the lips, uses the bended corner as the end point. This makes the calculated radius become slightly less than the actual radius. A modified code was made to consider this thing.

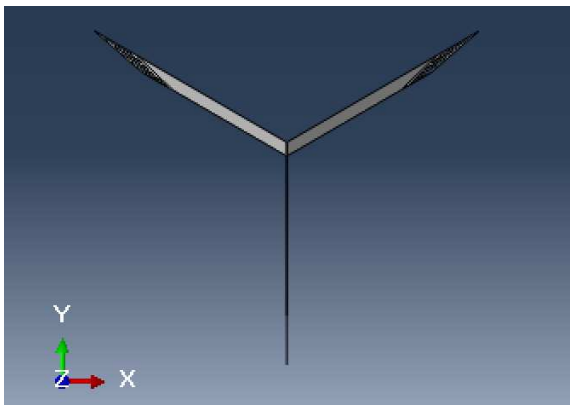


Figure 54. Created gusset part

Thicknesses of gusset plate is locked based on gusset plate-to-sector thickness ratio (*t_ratio*), set at value of 1.2.

The script sequence for gusset part inside the looping function is written below.

```
# -Profile sketch for gusset
# -Sketch gusset lines
# First point of the first sector
# Draw lines for the sketch of the gusset plate between 0, 0 and the
# -Extrude gusset part
```

Length of gusset plate is set equal to the diameter of assembly section, $l_g = d$. The profile of gusset was then extruded based on the above parameter.

The detail script for creating gusset part can be seen in Annex A.4 line 200 – 253.

- ***Perforation***

In part module, calculation of bolts position was also performed. Bolts, located both at connection part and along chord region, were modelled by holes with diameter equal to washer diameter. This was made to take into account the effect of contact interaction between the member plate and bolt washer. Creating holes were chosen instead of creating circle partition since the area inside the circle is of no interest in this study and also this method will obviate complicated mesh of circle region. Nonetheless, partitions still need to be made near the hole to apply structured mesh.

The connections provided by bolts will be modeled by tie constraining the perimeter of the holes. This will be further explained in the section of interaction module.

Workflow of the modeling process in Python for perforation of lips section can be described as follows.

1. Import length calculation from database pickle file
2. Calculate the gusset extrusion length, $\boldsymbol{l_1}$
3. Calculate the total extrusion length, $\boldsymbol{l_o = (l + l_1)m + l_1}$
4. Calculate the bolt spacing (s) as α ratio of the diameter, $\boldsymbol{s = b \,.\, d}$; $\boldsymbol{b \in (0.5, 1.0, 1.5)}$
5. Calculate the number of bolts (n) and the remainder distance to the edge (s_o)
   In Python use: "(n,s_o) = divmod (l,s)"
6. Create a list of z coordinates for the bolts;
   - First space: $\boldsymbol{s_1 = l_1 + \frac{s_o + s}{2}}$
   - All the following space $\boldsymbol{s_i = s_{i-1} + s}$
7. Create the holes
   - Holes are created using hole extrusion tool, require edges
   - The .getClosest method could be used to find the edges of the lips
8. Make longitudinal partitioning based on the calculated z-distance (± distance from the hole center).
   The longitudinal partitioning is needed to make structured mesh apart of the circle region. This partitioning can be done by:
   - Creating datum planes as offsets from principal plane *XY*
   - Partitioning face (select all faces) based on the datum planes

M16 bolts, with washer diameter d_w= 30mm, were used in all models. The choice of bolt size was made considering that the bolts will only act as a tie of the plates, without significant forces work on it.

Calculation of z-distances along *z*-axis is needed to get the exact position of bolts. This was done by calculating the bolt space along the clear span between joints and the remaining distance distribute evenly at both ends. Bolt space on connection region was made denser than the clear span to provide sufficient stiffness of the connection between chord and another member.

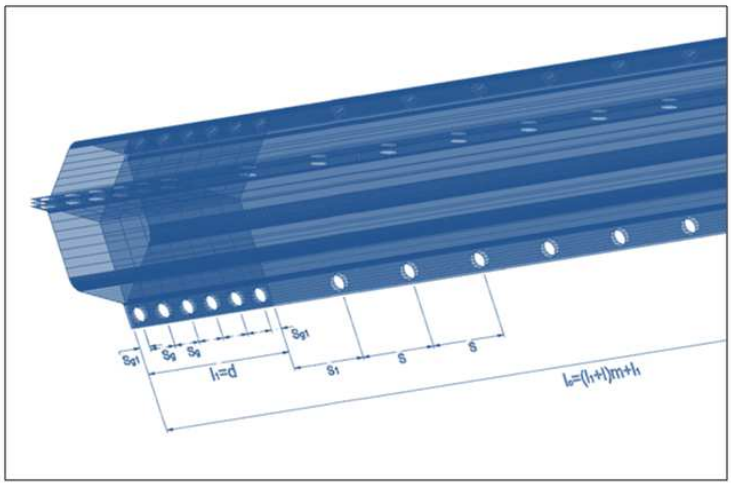


Figure 55. Scheme for bolt spacing

Script for calculation of bolt position is written below.

```
# Loop through the different bolt spacings (temporary b to change)
b = [3, 4, 5]
for m in range(1):
      # Calculate bolt positions
      # -Distance on the width
      bolts_w = current_llip/2
      # -Distances on the length
      current_b =  b[m]
      s = current_b*current_d
      (n0, s0) = divmod(current_l, s)
      s1 = (s0 + s)/2

      bolts_z1 = np.concatenate([[bolts_w], bolts_w + ((current_d -
      current_llip)/5) * np.linspace(1, 4, 4), [current_d - bolts_w]])
      bolts_z2 = np.concatenate([[current_d + s1], s1 + current_d + (s
      *np.linspace(1, n0-1, n0-1))])
      bolts_z3 = bolts_z1 + (current_l + current_d)
      bolts_z4 = bolts_z2 + (current_l + current_d)
      bolts_z5 = bolts_z3 + (current_l + current_d)

      bolts_z = np.concatenate([bolts_z1, bolts_z2, bolts_z3,
      bolts_z4,bolts_z5])
```

The coding for perforation adopts the tool for making holes extrusion available in ABAQUS. This tool uses edges to define position of holes. Two edges were needed, both are on the lip region. In order to select the edge, the .getClosest method can be used. Format of method:

`.getClosest(coordinates=((x,y,z,))`

This method will find the edge close to the selected point which taken from profile database. After that, by using the calculated distance from each edge, hole extrusion will be created. Distance1 was measured from edge1, equal to *z*-distance while distance2 was measure from edge2, equal to lipwidth/2.

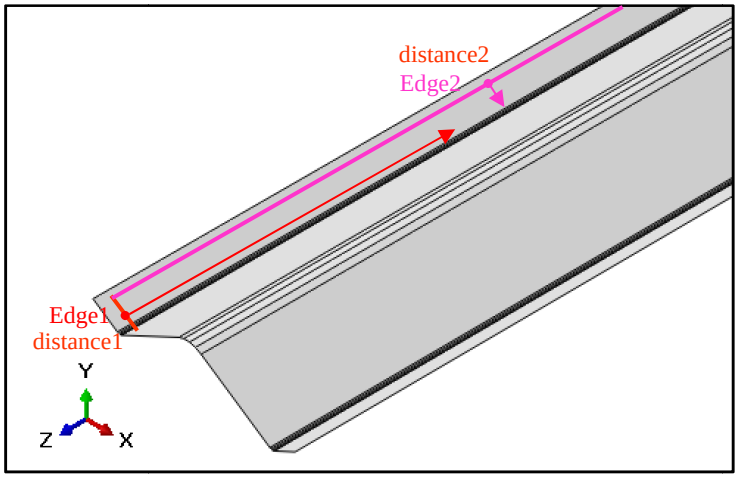


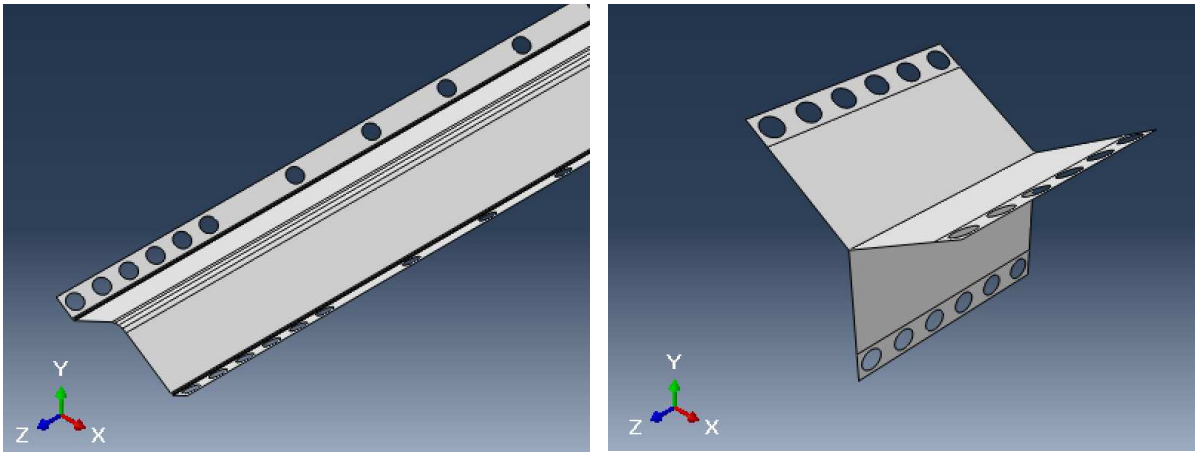
Figure 56. Method for perforation using edges in ABAQUS

The width of the lip as a function of lip-to-diameter ratio was made to take into account sufficient clearance for the bolt, which in this study uses M16 bolt. The part of script for hole extrusion is shown below.

```
# Washer diameter
d_washer = 30
# Initiate list to store datum planes
datum_p=[]

# Make holes
for o in range(int(bolts_z.shape[0])):
    sector_part.HoleBlindFromEdges(
      depth=1.0,
      diameter=d_washer,
      distance1=bolts_z[o],
      distance2=bolts_w,
      edge1=sector_part.edges.getClosest(coordinates=((profiles[i][j][k][0][
      1], profiles[i][j][k][1][1], 0),))[0][0],
      edge2=sector_part.edges.getClosest(coordinates=((profiles[i][j][k][0][
      0], profiles[i][j][k][1][0], 1),))[0][0],
      plane=sector_part.faces.getClosest(coordinates=((profiles[i][j][k][0][
      0], profiles[i][j][k][1][0], 0),))[0][0],
      planeSide=SIDE1)
```

The detail script for creating perforation, including bolt distance calculation and hole extrusion, can be seen in Annex A.4 line 92 – 157.



(a) Perforated 'sector' part (b) Perforated 'gusset' part

Figure 57. Modelling of perforation on part (Model 1-1-1-1)

3.4.3.3 Partition

Because of the perforation with circular shape there is issue with meshing. Therefore, the part will be partitioned into several parts. It is also advantageous for applying different mesh density at connection regions. Partition will be done in part level since dependent part instance type was chosen. The partition process can be presented in following script and figures.

- ***Sector***

Partitions were made at the vicinity of the holes as a strip with the help of series of datum planes. This method will allow the application of structured mesh in the region outside the holes, while keeping free mesh near the hole circle.

The position of datum planes was calculated by utilizing the bolts position points, taking distance half lip width toward both sides from center of hole.

```
# Create datum planes to be used for partitioning the sector
sector_part.DatumPlaneByPrincipalPlane(
      offset=bolts_z[o]-bolts_w,
      principalPlane=XYPLANE)
sector_part.DatumPlaneByPrincipalPlane(
      offset=bolts_z[o]+bolts_w,
      principalPlane=XYPLANE)
```

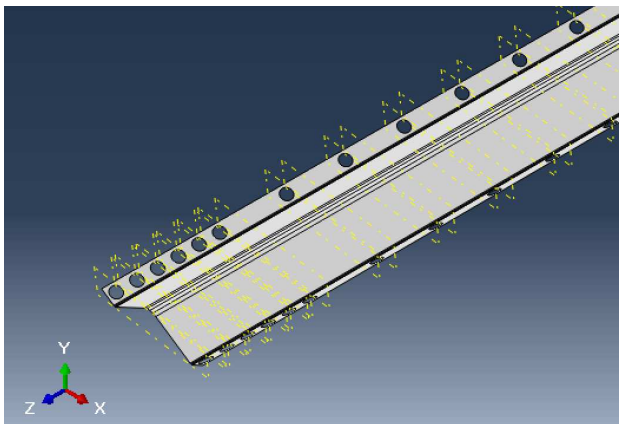


Figure 58. Datum planes for partitioning on sector part

The partitioning process adopting *.PartitionFaceByDatumPlane* method available in ABAQUS. It will partition the selected face, which is all face in this case.

```
# Partition the sector
# -Number of datum planes
n_dat = int(len(sector_part.datums))

# cut all the faces using the datum planes
for o in range((n_dat-2)/3):
      sector_part.PartitionFaceByDatumPlane(
      datumPlane=sector_part.datums.items()[o+1][1],
      faces=sector_part.faces[:])
```

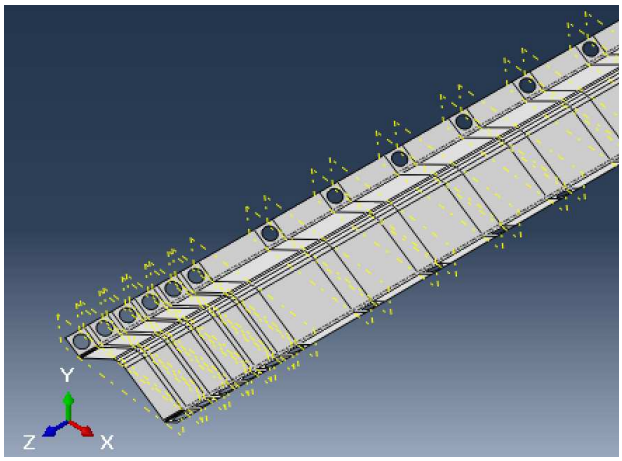


Figure 59. Partitioned sector part

- ***Gusset***

Partitioning of gusset was performed in the similar way with the partitioning of sector. However, for gusset *.PartitionFaceByShortestPath* method was adopted instead of *ByDatumPlanes* since the partition needed for gusset are along the longitudinal direction. Format of the method:
.PartitionFaceByShortestPath(faces=, point1=, point2=)

This method uses shortest path between two points as the cut line. First, datum points were set at the end sides of each fin of gusset plate at a distance of lip-width from the gusset tip. Coordinate of datum point was taken from profiles meta database. Selection of faces used *.getClosest* method to find the face from the coordinate of datum point.

After creating datum points at each fin of gusset plate, partition can be defined at the path connected those datum points. The *.getClosest* method was used to find the created datum points in partitioning process.

```
# Partition gusset

gusset_part.DatumPointByCoordinate((gp1[0]-current_llip*cos(5*pi/6),
gp1[1]-current_llip*sin(5*pi/6), 0),)
gusset_part.DatumPointByCoordinate((gp1[0]-current_llip*cos(5*pi/6),
gp1[1]-current_llip*sin(5*pi/6), current_d),)

gusset_part.PartitionFaceByShortestPath(
      faces=gusset_part.faces.getClosest(coordinates=((gp1[0], gp1[1],
      0),))[0][0],
      point1=gusset_part.datum.items()[0][1],
      point2=gusset_part.datum.items()[1][1],)
```

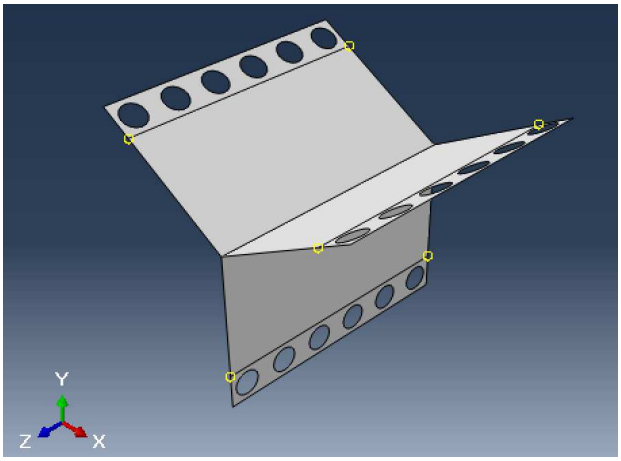


Figure 60. Datum points definition and partition of gusset part

The detail script for creating partition on sector part can be seen in Annex A.4 line 173 – 195, while for gusset part on line 299 – 342.

3.4.3.4 Property Module

3.4.3.4.1 Material Properties

The created part then will be assigned in the property module with material properties. For batch of model intended for buckling analysis, the elastic material properties are used. Modulus of elasticity, E = $210x10^3$ MPa and poisson ratio, v=0.3 was set in the script.

```
# Material
c_model.Material(name='pure-elastic')
c_model.materials['pure-elastic'].Elastic(table=((210000.0, 0.3), ))
```

For batch of model intended for static Riks analysis, the plastic material properties are added to the material assignment. This was done to understand the behaviour of the model in non-linear range when undergoing plastic stress. Ultimate resistance, load–displacement relationship, and post-yielding characteristic of the model will be observed. The value of this property was taken from stress-strain relationship in Table 7.

3.4.3.5 Section Module

As mentioned before, shell element will be assigned for all parts of the model. Thickness of the shell, *current_t*, is set based on the calculation of profile thickness stored in profiles meta database, as function of diameter-to-thickness ratio (*slend*).

```
# Create sections
# -for sector
c_model.HomogeneousShellSection(
      idealization=NO_IDEALIZATION,integrationRule=SIMPSON,
      material='pure-elastic', name='sector', numIntPts=5,
      poissonDefinition=DEFAULT, preIntegrate=OFF,
      temperature=GRADIENT, thickness=current_t,
      thicknessField='', thicknessModulus=None,
      thicknessType=UNIFORM, useDensity=OFF)
```

For gusset plate, similar section definition code was assigned. Only thickness of the shell is different, *current_tg*, which was set as function of gusset thickness-to-sector thickness ratio (*t_ratio*).

Section assignment was applied with middle surface as the offset type. For selection of part to be assigned, set of all faces was used.

```
# Assign sections
# -for sector
sector_part.Set(
      faces=sector_part.faces[:],
      name='AllSectorFaces')
sector_part.SectionAssignment(
      offset=0.0, offsetField='', offsetType=MIDDLE_SURFACE,
      region=sector_part.sets['AllSectorFaces'],
sectionName='sector',thicknessAssignment=FROM_SECTION)
```

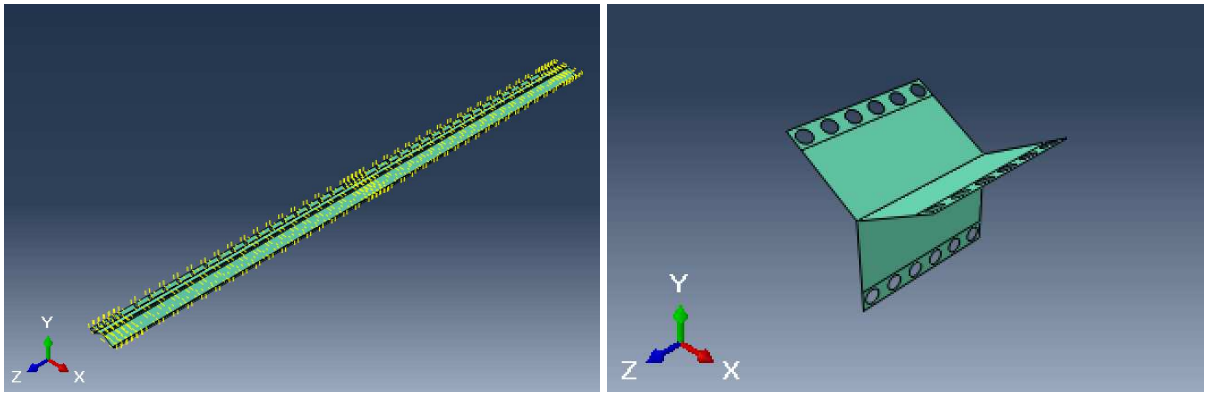


Figure 61. Section assigned to sector and gusset

3.4.3.6 Assembly Module

The modelled part was brought into the assembly module as a dependent part instance. It is the default type of part instance in ABAQUS/CAE. A dependent instance is only a pointer to the original part. In effect, a dependent instance shares the geometry and the mesh of the original part. As a result, it is possible to mesh the original part, but not on instances assembly. When the original part is modified, ABAQUS/CAE applies the same modification to all dependent instances of the part. Most modifications are not allowed on a dependent part instance. However, operations that do not modify the mesh and geometry of a dependent part instance are still allowed. Therefore, modification for mesh and geometry has to be done within the part and mesh module, respectively.

The advantages of dependent part instances are that they consume fewer memory resources and the parts need to be meshed only once and all the other dependent instances in the assembly module will be meshed accordingly. In addition, ABAQUS/CAE instances a dependent part instance in the input file by writing a single set of nodal coordinates and element connectivity to define the part along with a transform to define each part instance. Since the core of this parametric study is the script that will be run simultaneously on cluster systems and the most heavy duty part is the meshing so that to mesh one part on each model rather than three individual identical parts is preferred.

```
# Create assembly
c_assembly=c_model.rootAssembly
c_assembly.DatumCsysByDefault(CARTESIAN)
# -Sectors
```

```
c_assembly.DatumAxisByPrincipalAxis(
      principalAxis=ZAXIS)
c_assembly.RadialInstancePattern(
      axis=(0.0, 0.0, 1.0),
      instanceList=('sector-1', ),
      number=3, point=(0.0, 0.0, 0.0),
      totalAngle=240.0)
```

In order to have the assembly of one segment model, three instances of sector part and three instances of gusset part should be created. Sector instances were positioned through adopting radial pattern method available in ABAQUS. It allows the created instances going on according to the global coordinate system and positioned dependently to each other. The created instances were set refer to axis of rotation by 120˚. A datum axis was created at *z*-axis for this purpose.

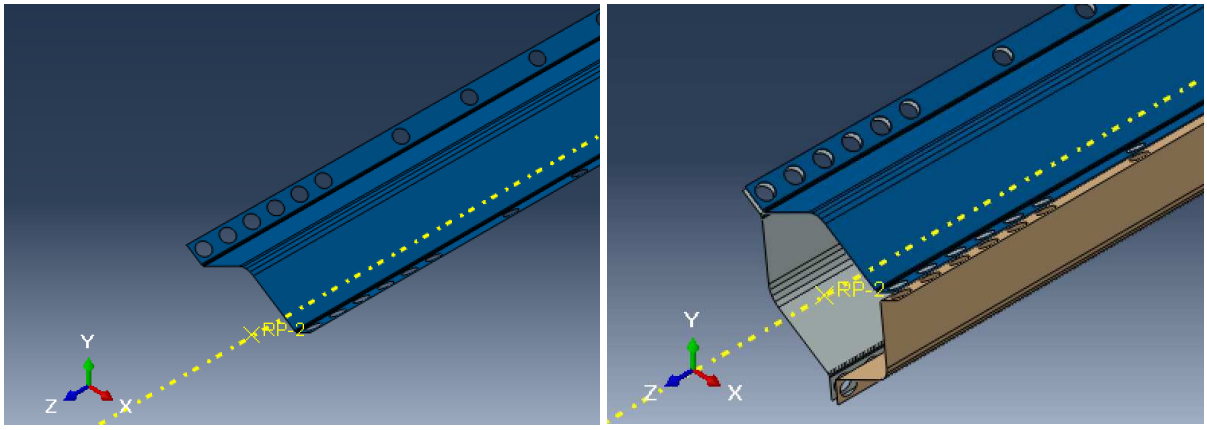


Figure 62. Sector instances

After gusset instances were created, they were going to be placed at connection region, at the far end and at the middle of segment model. Gusset instances were positioned through translation method with defined vector coordinate (0.0, 0.0, *z*). *Z*-point coordinate was taken from profiles metadata file.

```
# -Gusset plate
# --Create the instances
g1_instance=c_assembly.Instance(
      dependent=ON,
      name='gusset-1',
      part=gusset_part)
```

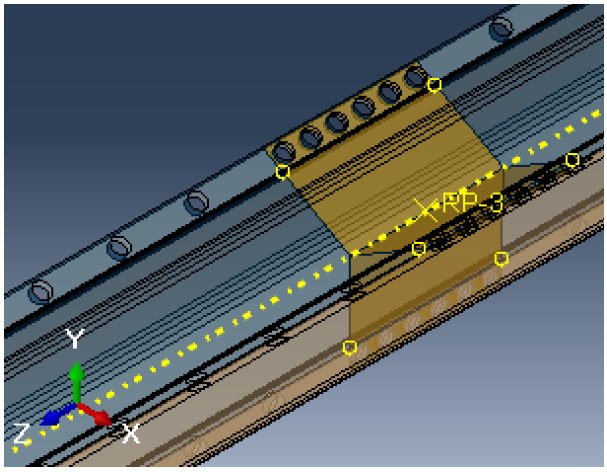


Figure 63.Gusset instances translation

```
# --Translate them to the right position
g2_instance.translate(vector=(
      0.0, 0.0, (current_l + current_d)))
g3_instance.translate(vector=(
      0.0, 0.0, 2*(current_l + current_d)))
```

The final geometry of assembled model is shown in Figure 64 and 65 below.

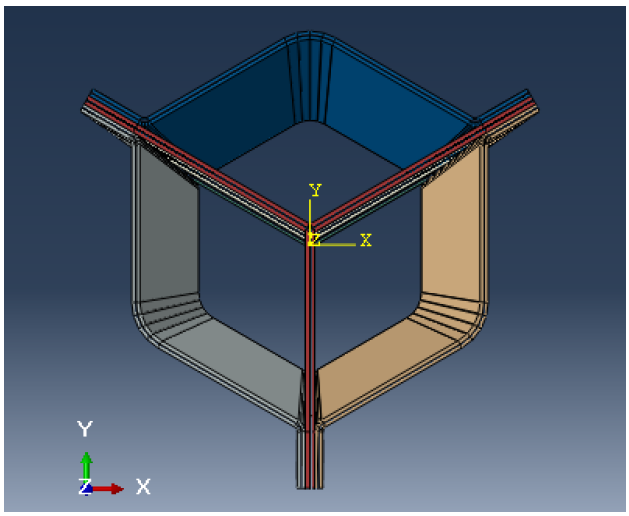


Figure 64. Extruded view of cross-section

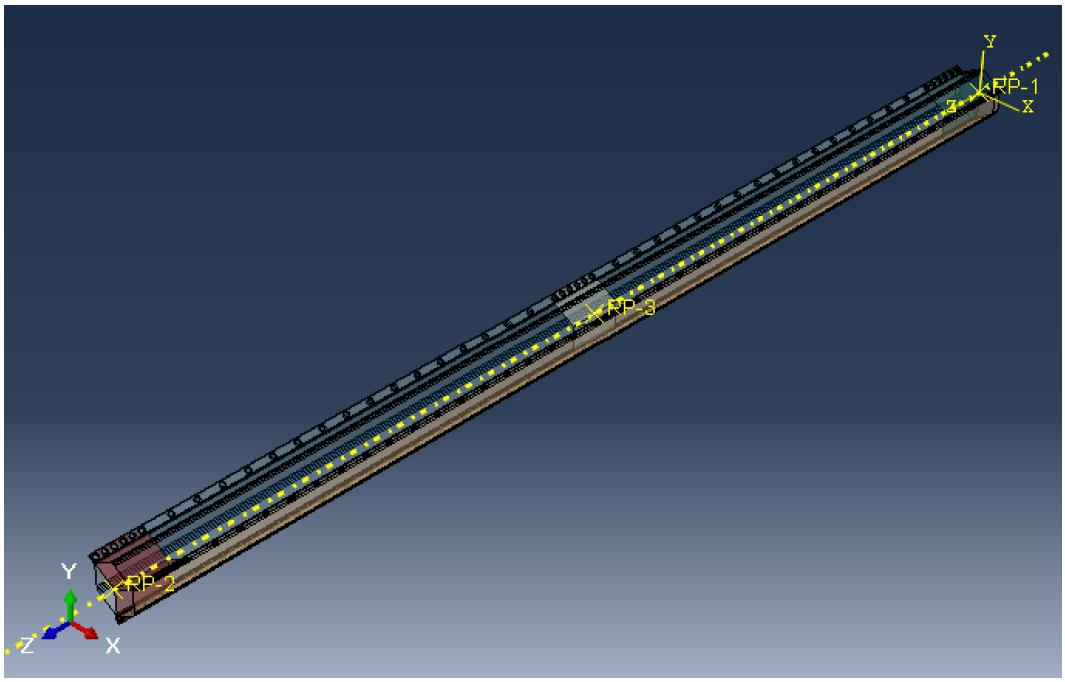


Figure 65. Final geometry of the assembly

3.4.3.7 Interaction Module

In the interaction module, there are several options for constraints and connections. Constraints defined in the interaction module will model constraints on the analysis degrees of freedom; partially or fully eliminate degrees of freedom of a group of nodes and couple their motion to the motion of a master node (or nodes).

In this modelling two constraint types were used, coupling constraint and rigid body constraint.

3.4.3.7.1 Coupling Constraint

Coupling interactions provide a constraint between a reference node and the nodes on a surface (the coupling nodes). It will limit the motion of a group of nodes to the rigid body motion defined by a single node (reference node).

In the model, this type of constraint was applied for applying loads and boundary conditions to a model and model the end conditions, as well. Kinematic coupling constraint which constraining all six degree of freedoms (u1, u2, u3, ur1, ur2, ur3) was used in the modeling.

Coupled elements:

- Reference point for BC and load application
- End chord including gusset

The reference point (RP) end node as master node was coupled with the front edge of the end chord as slave nodes, as shown in Figure 66.

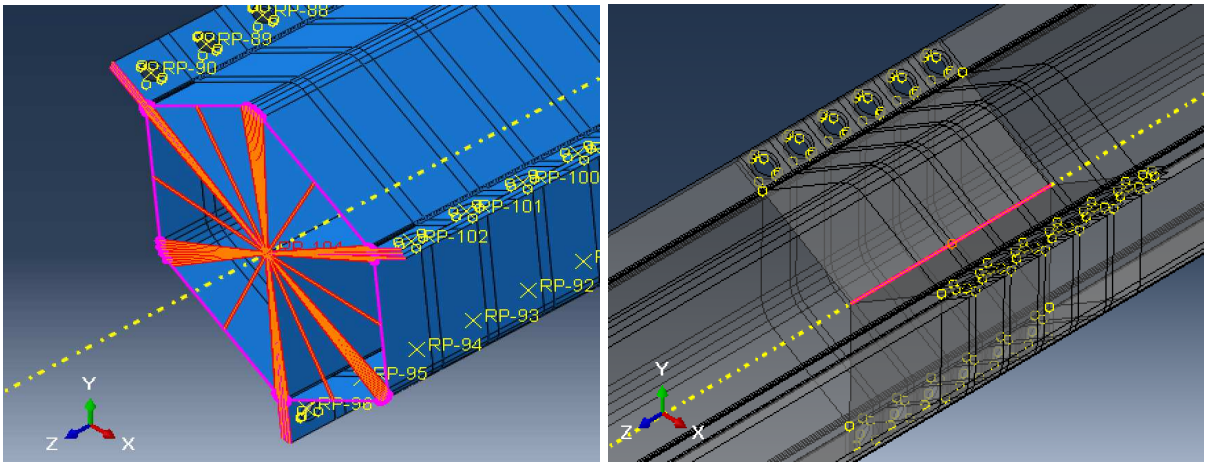


Figure 66. Kinematic coupling constraints

First the reference points (RPs) were created. These reference points will also be used for boundary conditions (BCs) and loading conditions. RPs were made at the two ends and at the middle of the column.

```
# Create reference points
# -RPs for the faces at the two ends of the columns
c_assembly.ReferencePoint((0.0, 0.0, 0.0))
c_assembly.ReferencePoint((0.0, 0.0, (2*current_l + 3*current_d)))

# - RP at the middle
c_assembly.ReferencePoint((0.0, 0.0, (current_l + 1.5*current_d)))
```

In this stage, it is important to create sets of geometry since the interaction will involve numbers of nodes and edges. Identification of set will ease the workflow of coding.
Sets then made for each master and slave node, the 'RP-end1 set' and 'RP-end2 set' for the reference point, and 'end1-face' and 'end2-face' for the end edges. The creation of coupling constraint used *.findAt* method to select the reference point (RP) as master node, and *.getByBoundingBox* method to select the faces of slave nodes. The arguments to *findAt* are an arbitrary point on an edge, face, or cell or the X-, Y-, and Z-coordinates of a vertex. *findAt* returns an object that contains the id of the vertex or the id of the edge, face, or cell that includes the arbitrary point The *.getByBoundingBox*method is capable of selecting objects that lie entirely within the selection volume. Six points need to be defined for the box. Format of the method:

```
.findAt(x, y, z)
.getByBoundingBox(xmin, ymin, zmin, xmax, ymax, zmax)
```

```
# - End face couplings to reference points
# End 1
c_assembly.Set(
      name='RP-1-set',
      referencePoints=(c_assembly.referencePoints.findAt((0, 0, 0)),
))
c_assembly.Set(
      edges=g1_instance.edges.getByBoundingBox(-current_d,-
      current_d,0,current_d,current_d,0),
      name='end1-face',)
c_model.Coupling(
      controlPoint=c_assembly.sets['RP-1-set'],
      couplingType=KINEMATIC, influenceRadius=WHOLE_SURFACE,
      localCsys=None, name='end1-coupling',
      surface=c_assembly.sets['end1-face'],
      u1=ON, u2=ON, u3=ON, ur1=ON, ur2=ON, ur3=ON)
```

The second end coupling at another end (End 2) and the third coupling at the middle connection (Middle) were made in the same way as above. A distinction for middle connection is that the mid-RP master nodes coupled with the three fins' intersection edge as the slave nodes (Figure 66).

3.4.3.7.2 Tie Constraint

Tie constraint ties two separate surfaces together so that there is no relative motion between them. Hence, the two tied surfaces expected to behave as rigid body. This type of constraint allows fuse together two regions even though the meshes created on the surfaces of the regions may be dissimilar. This type of constraint was tried to tie the bolt holes along the sector plates and gusset plates. Modelling of bolt connection using tie constraint was performed in consideration that the rigidity of bolts are much higher compared to the connected plates, so that by constraining the perimeter of holes as rigid body will represent the bolt connections on the model.

Tied element:

- Holes perimeter on sector and gusset plates

The master nodes are nodes along the holes perimeter and the slave nodes are the nodes along the holes perimeter at the opposite side.

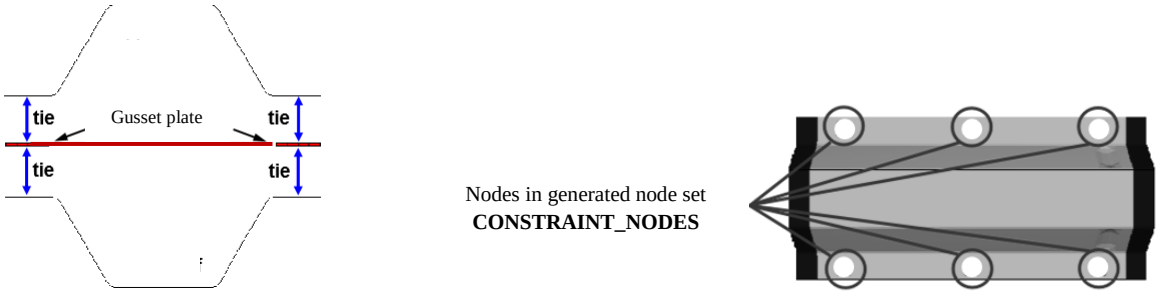


Figure 67. Tie constraint of holes perimeter

The codes for tie constraint were made by first creating the sets of node region for each constraint pair. Sets of node were stored to later facilitate nodes selection using *.findAt* method.

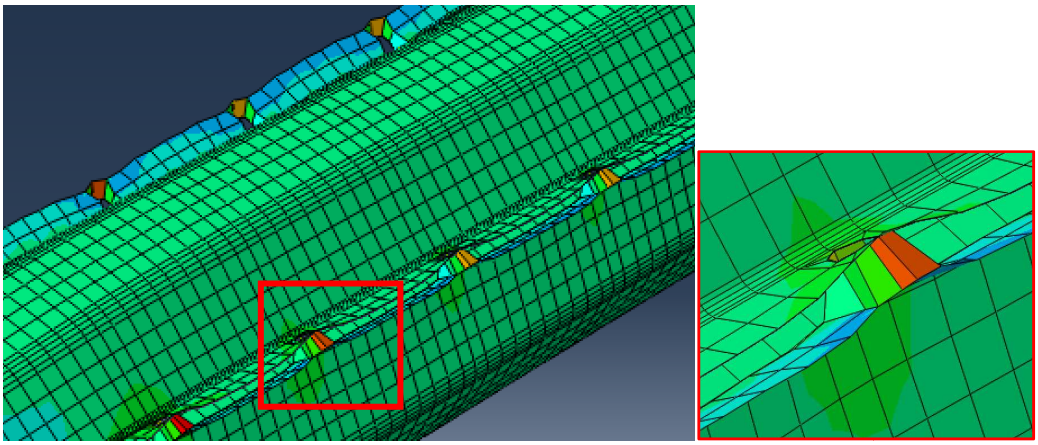


Figure 68. Result of static analysis at tie constraint region

As it can be noticed in Figure 68, the tie constraint physically displaces the nodes of the slave region to meet the master nodes, which is unrealistic.

The type of constraint intended in this study is the one that only couples the node DOFs, without changing the geometry. The attempt to constraint the connection holes then will be done with rigid body type of constraint instead of tie. For this type of constraint, reference points, RPs, must be created for each bolt and then all corresponding surfaces by coupled as a rigid body to the RP.

3.4.3.7.3 Rigid Body Constraint

A rigid body constraint allows constrain the motion of regions of the assembly to the motion of a reference point. The relative positions of the regions that are part of the rigid body remain constant throughout the analysis.

Codes for rigid body constraint principally made in the same way with the tie constraint previously described, only sets of reference points need to be created to act as master nodes. Sets of node were stored as variables, to facilitate nodes selection using *.findAt* method.

First, position of each bolt was defined for sector (‘sh’) and gusset (‘gh’) using profiles database and function of bolt position in gusset plate, respectively.

```
# Position of the holes on the cross-section
sh11 = np.array([profiles[i][j][k][0][1], profiles[i][j][k][1][1]])
sh12 = np.array([profiles[i][j][k][0][-2], profiles[i][j][k][1][-
2]])
gh1 = (gp1[0]-bolts_w*cos(5*pi/6), gp1[1]-bolts_w*sin(5*pi/6))
gh2 = (gp2[0]-bolts_w*cos(-pi/2), gp2[1]-bolts_w*sin(-pi/2))
gh3 = (gp3[0]-bolts_w*cos(pi/6), gp3[1]-bolts_w*sin(pi/6))
```

These sets of point then multiplied in order to obtain the points on the other instances. Rotation matrix (*Rmat*) method was used to calculate the points for sector plate.

```
# Rotation matrix to get the points of the other 2 instances
Rmat = np.array([[cos(-2*pi/3), -sin(-2*pi/3)], [sin(-2*pi/3), cos(-
2*pi/3)]])

# Calculate the points by multiplying with the 120 degrees rotation
matrix
sh21 = sh11.dot(Rmat)
sh22 = sh12.dot(Rmat)
sh31 = sh21.dot(Rmat)
sh32 = sh22.dot(Rmat)
sh = ((sh11, sh12), (sh21, sh22), (sh31, sh32))
```

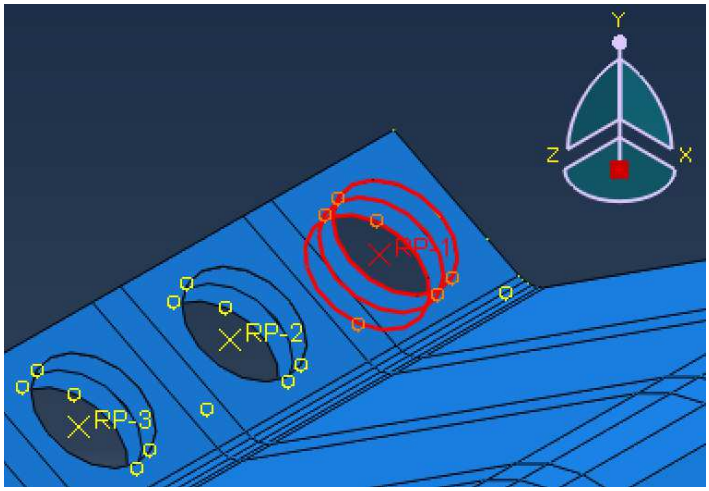


Figure 69. Rigid body constraint assignment at 'bolt-1'

After that, reference points and set of reference point were created by using *.findAt* method. In the following section will be described the codes for rigid body constraint as End1 connection. The remaining region, i.e., Middle connection, Span1, Span2, and End2 connection were made with the same flow.

The RPs and tie region were made in loop of number of bolts along z-direction based on bolt function previously defined. *findAt* method used to select 'sh' and 'gh' points and stored them in set. This function will create constraint between the set of reference points and the corresponding set of tie regions, with number of constraints as many as the hole pairs.

```
# Create reference points for the bolt rigid body couplings
# Create the necessary sets and the constraints for all the bolts
# End 1 connection
for oo in (range(3)):
      ii=1
      for o in tuple(bolts_z1):
            c_assembly.ReferencePoint((gh[oo-3][0], gh[oo-3][1],
            float(o)))
            c_assembly.Set(
                  edges=s_instance[oo-3].edges.findAt(((sh[oo-
                  3][0][0],sh[oo-3][0][1], float(o)-d_washer/2), ),
                  )+\
                  s_instance[oo-2].edges.findAt(((sh[oo-2][1][0],
                  sh[oo-2][1][1], float(o)-d_washer/2), ), )+\
                  g1_instance.edges.findAt(((gh[oo-3][0], gh[oo-3][1],
                  float(o)-d_washer/2), ), ),
                  name='b'+str(ii)+str(oo)+'set1')
            c_model.RigidBody(
                  name='b1'+str(ii)+str(oo)+'joint1',
                  refPointRegion=Region(referencePoints=(c_assembly.re
                  ferencePoints.findAt((gh[oo-3][0], gh[oo-3][1],
                  float(o))), )),
                  tieRegion=c_assembly.sets['b'+str(ii)+str(oo)+'set1'
                  ])
```

The detail script of interaction can be seen in Annex A.4 line 504 – 717. Result for the application of rigid body constraint is shown in Figure 70.

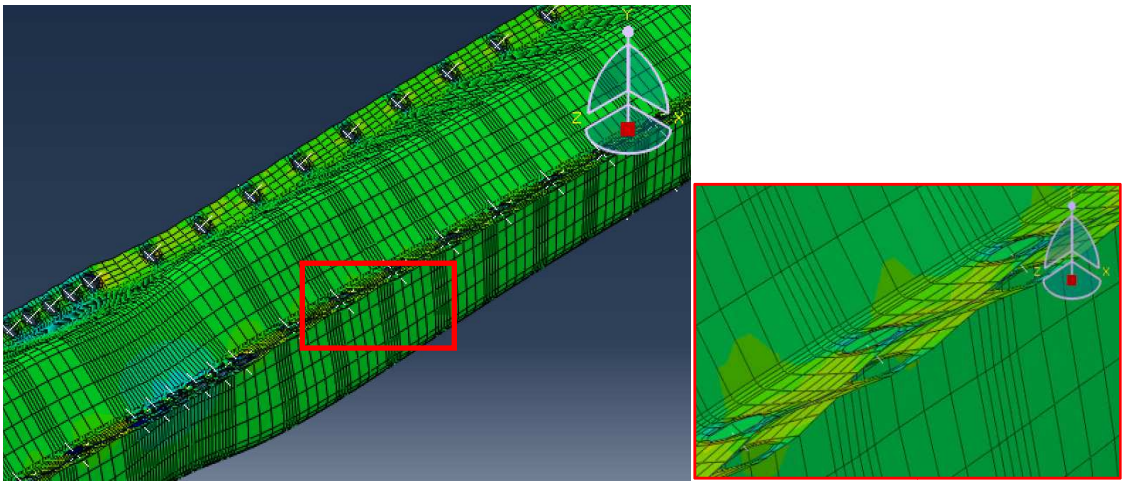


Figure 70. Result of static analysis at rigid body constraint region

It can be seen in the figure, the rigid body constraint works well, as expected in rigid connection. The geometry at hole circle does not change and tied accordingly. Therefore, this type of constraint was used for modeling the bolt connection.

3.4.3.8 Load Module

Boundary conditions have a profound impact on the structural response and therefore they must be modelled as closely as possible to the physical situation in practical thin-walled steel systems. Despite this, their numerical models should not be overly complicated so as to impair modelling efficiency and computational stability.

The boundary conditions adopted in the model correspond to those applies for truss structure. Pinned support at the two ends of the column was used. As mentioned above, the boundary condition will be put on the end reference point (RP) and connected to the member by kinematic coupling constraint. This way, the coupling nodes, i.e., the chord ends are constrained to the rigid body motion of a single node (RP). Therefore, all the constrained degrees of freedom transmitted at this node. It will allow the end chord to rotate globally instead of each edge move locally if BC along edges is applied.

BCend-1 at one end reference node has *u1*, *u2*, *u3* restrained translational DOFs and *ur3* restrained rotational DOFs, while BC end-2 at the other end has *u3* free to accommodate the application of load. A BC was added for the middle joint. This joint is stabilised by the converging diagonals so the global buckling length is the length of one span. In this middle joint BC, All DOFs of the nodes on the edges are coupled to the RP and the RP is restrained on U1 and U2. It can move on the z axis and it can rotate freely because the converging diagonals do not provide any rotational stiffness (it is very small and neglected).

```
# BCs
end1_BC=c_model.DisplacementBC(
      amplitude=UNSET, createStepName='Initial',
      distributionType=UNIFORM, fieldName='',
      localCsys=None, name='fix-end1',
      region=Region(referencePoints=(c_assembly.referencePoints.findAt
      ((0, 0, 0)), )),
      u1=SET, u2=SET, u3=SET, ur1=UNSET, ur2=UNSET, ur3=SET)
```

The RP was selected from the predefined set of reference points by using *.findAt* method. Script for end2-BC and middle-BC were made in the similar way as above. All boundary conditions were created in the initial step.

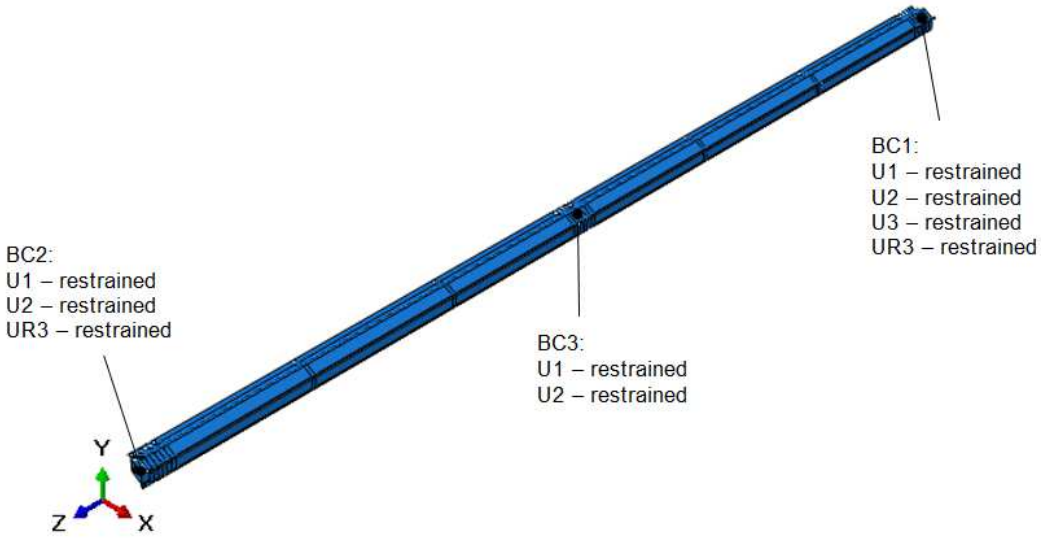


Figure 71. Boundary condition of the model

As for loading condition, two loading conditions were set.

1. Buckling analysis: concentrated force of 1N was applied at end2-BC
2. Riks analysis:
   a. Riks_N (axial only): Concentrated force of 510 *x* area of cross section at end2-BC. 510MPa is the nominal ultimate stress for S355. It is used to estimate a load as the highest load that could be applied on the cross-section. So the resistance is definitely going to be lower than that in all cases
   b. Riks_NM (axial and bending): Concentrated force of 510 *x* area of cross section at end2-BC and 5, 10, 15% bending moment resistance of the cross-section at mid-BC. This value is chosen since the presence of bending moment in truss structure actually not significant. The bending moment is induced by forces from diagonals.

```
# Apply concentrated force
N_pl_rd = 510*area

# Apply bending moment at the mid-connection
# Calculate the magnitude of moment as 10% of moment resistance
W = current_Iy/(current_d/2)
M_resist = W*current_fy
M = 0.1*M_resist

r_model_NM.Moment(
 cm1=-M,
 createStepName='RIKS',
 distributionType=UNIFORM,
 field='',
 localCsys=None,
 name='moment',
 region=c_assembly.sets['RP-Mid-set'])
```

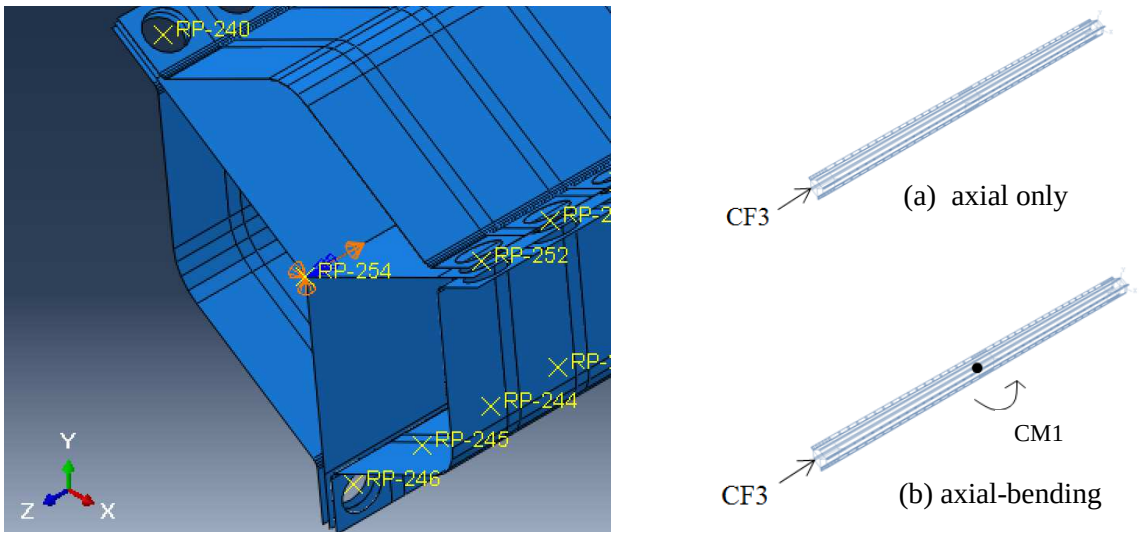


Figure 72. Boundary conditions and loading conditions

3.4.3.9 Mesh Module

It is important to note, that appropriate element type should be selected in the meshing module, in order to provide ABAQUS with information about the interpolation it should apply to those elements. For thin-walled structures it is widely acknowledged that shell elements are the most suitable type for FE modelling because it is able to capture local, distortional, as well as localised buckling of the structure so that a realistic response can be predicted. ABAQUS offers a dozen commonly used shell elements for 3-D stress/displacement analysis, such as S3, S3R, S4, S4R, S4R5, S8R, S8R5, S9R5, etc., of which S4R, as shown in Figure73, was chosen for this study.

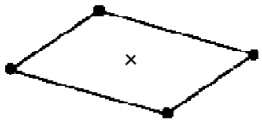


Figure 73.Illustration of ABAQUS shell element S4R

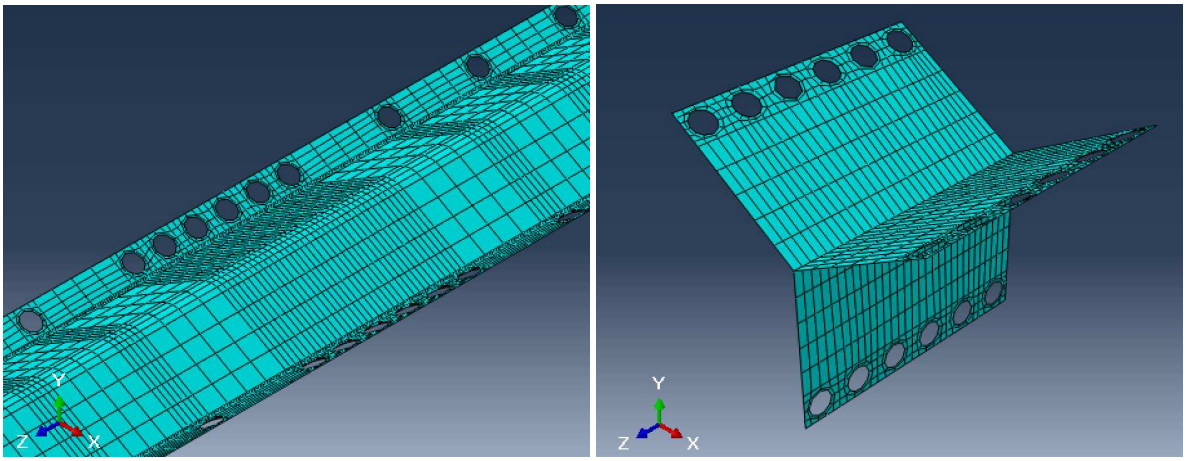
Element S4R is a standard large-strain shell element with "4-node general-purpose shell, finite membrane strains" (ABAQUS 6.14 Documentation). It has six degrees of freedom per node: three translations and three rotations. It also has a uniformly reduced integration to avoid shear and membrane locking. It also converges to a shear flexible theory for thick shells and to a classical theory for thin shells. Hence, it can be concluded that the element chosen is a suitable element for the buckling analysis of thin-walled structures.

As already mentioned in the assembly module, all part instances in the model are dependent type. It means that meshing of those elements should be performed for each part instead of in assembly level. The mesh module will associate the mesh from part to all dependent instances. In order to keep the meshed element as close to square as possible and limit the free mesh only to region right at the peripheral of the hole circle, partitions were created as described in the part module.

All of the shell elements were seeded by size of seeds. An adequate mesh of the structures is very important when dealing with finite element method analysis. If a mesh is too coarse, the results obtained from the analysis may not be sufficiently accurate. On the other hand, if a mesh is too refined, the analysis can develop accurate results but it will increase the processing time with computational costs. In order to have the accurate results without spending too much processing time, it was decided to model all the structures with a reasonable mesh. After running several convergence studies, the mesh with approximate size of 30 was chosen.

```
# Global seeding in mm
seedsize = 30
# -Sector
sector_part.setMeshControls(algorithm=MEDIAL_AXIS,elemShape=QUAD,
      regions=sector_part.faces[:])
 sector_part.seedPart(deviationFactor=0.1, minSizeFactor=0.1,
       size=seedsize)
sector_part.generateMesh()
```

Mesh control with quadratic shape of element was used since it can be applied to any planar or curved surface. The reason behind this is that the high order approximation for the finite element (keeping the same size) leads to the small error for the solution if all parameters (boundary conditions, geometry, materials) are sufficiently smooth. Moreover, it can avoid error due to shear locking in bending problem commonly found in triangular shape. Thus the quadratic approximation is used.



Sector part Gusset part

Figure 74. Meshed part

3.4.3.10Step Module

In ABAQUS the user divides the complete load history of the simulation into a number of steps. Each step is a period of "time," specified by the user, for which ABAQUS calculates the response of the model to a particular set of loads and boundary conditions. The user must specify the type of response, known as the analysis procedure, during each step and may change analysis procedures from step to step.

ABAQUS divides all of its analysis procedures into two main groups: linear perturbation and general. In this parametric study linear eigenvalue analysis and static Riks analysis were employed. Linear eigenvalue analysis is generally used to estimate the critical (bifurcation) load and buckling shape of the structure. On the other hand, load-deflection (Riks) analysis had been used to incorporate concern about material nonlinearity, geometric nonlinearity prior to buckling, and postbuckling response.

```
c_model.BuckleStep(maxIterations=300, name='Buckling', numEigen=4,
 previous='Initial', vectors=10)
```

The Riks method uses the load magnitude as an additional unknown; it solves simultaneously for loads and displacements. Therefore, another quantity must be used to measure the progress of the solution; ABAQUS/Standard uses the "arc length," l, along the static equilibrium path in load-displacement space. This approach provides solutions regardless of whether the response is stable or unstable.

```
# Create RIKS step
r_model_N.StaticRiksStep(name='RIKS', previous='Initial', nlgeom=ON,
 maxNumInc=30,initialArcInc=0.2)
```

Two kinds of Riks analysis were employed in this study, axial only (N) and axial bending (N+M) analysis.

3.4.3.10.1 Imperfection

The geometrically and materially nonlinear analysis with imperfection using Riks solver obtain the shape of imperfection according to the first buckling mode shape from linear perturbation analysis, and its magnitude according to recommendation in EC3-1-5.The procedure to include this imperfection is performed by editing keywords. The displacement data for different buckling shapes is written to the node file, which then is used for Riks analysis as the shape of imperfection. In Python the process of writing, deleting, and modifying keyword is possibly done with the built-in

keywordBlock method. This method automates the process to synchronize, insert, and replace text block.

```
# Define a method to get the block number of a specific string in
thekeywords
c_model.keywordBlock.synchVersions()
def GetBlockPosition(buckle_model,blockPrefix):
 pos = 0
 for block in c_model.keywordBlock.sieBlocks:
       if
string.lower(block[0:len(blockPrefix)])==string.lower(blockPrefix):
             return pos
       pos=pos+1
 return -1
```

The amplitude of imperfection taken for the analysis is equal to *s/2000*, with *s* is bolt spacing. Distance of bolt spacing was chosen instead of overall length since it was revealed that the dominant buckling mode occurred is sectional buckling, not the flexural one. The use of elastic buckling mode as the shape of geometric imperfections itself and the chosen magnitude was a conservative approach (Moen 2008) **[**30**]**.

```
# Change keywords to include initial imperfections
amp_impf = s/2000
r_model_N.keywordBlock.replace(GetBlockPosition(r_model_N, '*step')-1,
'\n** ----------------------------------------------------------------
\n** \n**********GEOMETRICAL IMPERFECTIONS\n*IMPERFECTION,FILE='
+ str(buckle_model) +',STEP=1\n1,'+ str(float(amp_impf))
+'\n2,'+str(float(amp_impf)) +'\n3,'+ str(float(amp_impf))
+'\n4,'+str(float(amp_impf)) +'\n**')
```

3.4.3.10.2 Field and History Output

In order to obtain intended output, in this module all required output data are specified. In field output request the following data was requested:

- All stress components ‘S’, mises equivalent stress ‘Mises’, all strain components ‘E’, effective plastic strain ‘PEEQ’, and all physical displacement components ‘U’

while in history output request:

- Reaction force RF3 at RP-1
- Displacement U3 at load application point RP-2
- Rotation UR1 at middle connection RP-Mid

3.4.3.11Job Module

Once all of the tasks involved in defining a model is finished, the Job module can be assigned to analyze the model. The job will be created for each model and then submitted to cluster which comprise multiple processor. It will divide the model analysis into multiple tasks according to number of processor.

At the end of script, a command to create input file was made. Therefore, the input file will be created for each model.

```
# Write the input file
mdb.jobs[riks_model_NM].writeInput()
```

The script for Job assignment can be seen in Annex A.4 line 860 – 884.

4 PARAMETRIC STUDIES OF THE PROPOSED STRUCTURAL MEMBERS

Chapter 3 detailed the development of the FE models, and showed that the input file coded in Python was able to generate reliable FE models of proposed polygonal thin-walled columns. These FE models can therefore be readily used for extensive parametric studies towards design purposes. A total of more than 150 FE elastic buckling and collapse analyses have been conducted in the parametric studies, involving variables such as number of corner (n), diameter of cross-section (d), slenderness (slend), yield strength (fy), bending radius to thickness ratio (rcoef), number of points along the bend (nbend), lip length to diameter ratio (l_ratio), gusset to sector thickness ratio (t_ratio), bolt spacing (b), and member length (l). Evaluation of the results will be carried out with Factorial Design.

4.1 Post-Processing

In this chapter, results from the analysis will be presented. After downloading all result files from the cluster, a method was used to call the required output variables and arrange them in a convenient format for the further study. This was done through automation using Python script dedicated to collect data from .odb files and present them in a certain tabulated database.

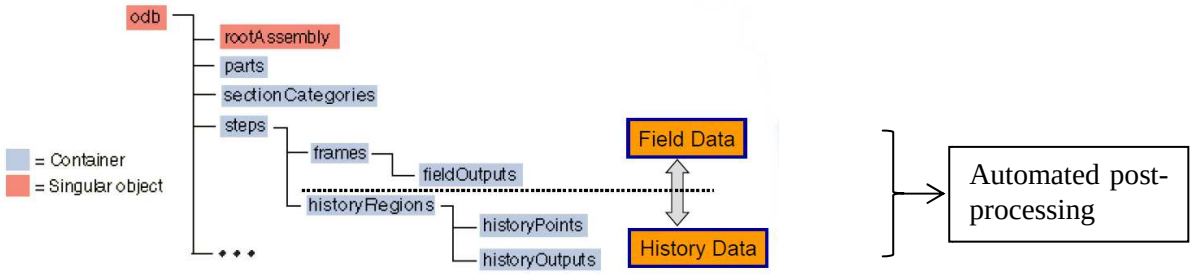


Figure 75. Abaqus result in .odb file

To access the output database, `odbAccess` module was used. This post-processing work will produce an external data outside ABAQUS, with the following approach:

- Open and read data from all .odb files
- Write the required data in predefined format to an output database (.dat)
- Capture the visualization from viewport module and save them in certain format (.png)

By doing so, data analysis, e.g. statistics and probability of the results can be carried out more handily. In this parametric study, the output database was analysed by statistical analysis, i.e. factorial design which will be described in the next section.

Complete script of the output reading automation, including .odb data extraction and visualization data import from the viewport module can be seen in Annex A.5.Example of result table extracted from .odb file using this automation is presented in Annex D.

4.2 Factorial Design

In this parametric study, results from FEA will be assessed with Factorial Experimental Design. In statistics, a full factorial experiment is an experimental design whose design consists of two or more factors, each with discrete possible values or "levels", and whose experimental units take on all possible combinations of these levels across all such factors. A full factorial design may also be called a fully crossed design. Such an experiment allows the investigator to study the effect of each factor on the response variable, as well as the effects of interactions between factors on the response variable. Moreover, multiple response variables can be studied at once.

For the vast majority of factorial experiments, each factor has only two levels. For example, with two factors each taking two levels, a factorial experiment would have four (2^k) treatment combinations in total, with *k* is number of factors.

By applying Factorial Design in FEA analysis, the following objectives are to be achieved:

1. Screen for important factors
2. Determine factor interactions
3. Systematically study how different modelling choices affect the FE model response.

The example of two level full factorial design with three factors: 2^3 design, is shown below.

Run	A	B	C	AB	AC	BC	ABC	y (response)
1	−	−	−	+	+	+	−	y_1
2	+	−	−	−	−	+	+	y_2
3	−	+	−	−	+	−	+	y_3
4	+	+	−	+	−	−	−	y_4
5	−	−	+	+	−	−	+	y_5
6	+	−	+	−	+	−	−	y_6
7	−	+	+	−	−	+	−	y_7
8	+	+	+	+	+	+	+	y_8

Factors are A, B, and C, with interaction AB, AC, BC, and ABC. Number of runs to be performed grows exponentially with number of factors (2^k). Response variable for each run as a function of interaction is indicated as $y_1 \ldots y_8$.Main and interaction effect then can be calculated, as follows.

$$A_{\text{main}} = \frac{(y_2 + y_4 + y_6 + y_8)}{4} - \frac{(y_1 + y_3 + y_5 + y_7)}{4}$$
$$B_{\text{main}} = \frac{(y_3 + y_4 + y_7 + y_8)}{4} - \frac{(y_1 + y_2 + y_5 + y_6)}{4}$$
$$AB_{\text{int.}} = \frac{(y_1 + y_3 + y_6 + y_8)}{4} - \frac{(y_2 + y_4 + y_5 + y_7)}{4}$$
$$ABC_{\text{int.}} = \frac{(y_2 + y_3 + y_5 + y_8)}{4} - \frac{(y_1 + y_4 + y_6 + y_7)}{4}$$

Or can be expressed:

$$\text{Main and interaction effect} = \bar{y}^{+} - \bar{y}^{-} \text{ (bar = mean value)}$$

Factors are cancelled out (for Factor A):

$$\begin{aligned}\bar{y}^{+} = (&A^{+} + B^{-} + C^{-} + AB^{-} + AC^{-} + BC^{+} + ABC^{+} \\ &+A^{+} + B^{+} + C^{-} + AB^{+} + AC^{-} + BC^{-} + ABC^{-} \\ &+A^{+} + B^{-} + C^{+} + AB^{-} + AC^{+} + BC^{-} + ABC^{-} \\ &+A^{+} + B^{+} + C^{+} + AB^{+} + AC^{+} + BC^{+} + ABC^{+})/4 \\ &= 4A^{+}/4 = A^{+} \\ \bar{y}^{-} = &\ldots = 4A^{-}/4 = A^{-}\end{aligned}$$

Computing main and interactions effects in matrix notation:

$$\boldsymbol{D} = \begin{bmatrix} -1 & -1 & -1 & +1 & +1 & +1 & -1 \\ +1 & -1 & -1 & -1 & -1 & +1 & +1 \\ -1 & +1 & -1 & -1 & +1 & -1 & +1 \\ +1 & +1 & -1 & +1 & -1 & -1 & -1 \\ -1 & -1 & +1 & +1 & -1 & -1 & +1 \\ +1 & -1 & +1 & -1 & +1 & -1 & -1 \\ -1 & +1 & +1 & -1 & -1 & +1 & -1 \\ +1 & +1 & +1 & +1 & +1 & +1 & +1 \end{bmatrix} = \text{design matrix}$$

$$\boldsymbol{R} = \begin{bmatrix} y_1 & y_2 & y_3 & y_4 & y_5 & y_6 & y_7 & y_8 \end{bmatrix}^{\mathrm{T}} = \text{response vector}$$

$$\boldsymbol{E} = \frac{2}{n_{\text{runs}}} \boldsymbol{D}^{\mathrm{T}} \boldsymbol{R} = \begin{bmatrix} A_{\text{main}} & B_{\text{main}} & C_{\text{main}} & AB_{\text{int.}} & AC_{\text{int.}} & BC_{\text{int.}} & ABC_{\text{int.}} \end{bmatrix}^{\mathrm{T}}$$

where n_{runs}= 8 is the total number of simulations/runs.
Hence,

$$\boldsymbol{E} = \frac{2}{n_{\text{runs}}} \underbrace{\begin{bmatrix} -1 & +1 & -1 & +1 & -1 & +1 & -1 & +1 \\ -1 & -1 & +1 & +1 & -1 & -1 & +1 & +1 \\ -1 & -1 & -1 & -1 & +1 & +1 & +1 & +1 \\ +1 & -1 & -1 & +1 & +1 & -1 & -1 & +1 \\ +1 & -1 & +1 & -1 & -1 & +1 & -1 & +1 \\ +1 & +1 & -1 & -1 & -1 & -1 & +1 & +1 \\ -1 & +1 & +1 & -1 & +1 & -1 & -1 & +1 \end{bmatrix}}_{\boldsymbol{D}^{\mathrm{T}}=7\times 8} \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \end{bmatrix}}_{\boldsymbol{R}} = \begin{bmatrix} A_{\text{main}} \\ B_{\text{main}} \\ C_{\text{main}} \\ AB_{\text{int.}} \\ AC_{\text{int.}} \\ BC_{\text{int.}} \\ ABC_{\text{int.}} \end{bmatrix}$$

The parametric study that performed in this thesis is a multi-level full factorial with four factors (diameter, cross section slenderness, member slenderness, and bolt spacing). To evaluate the result of a factorial design, a graph that is called factor interaction plot is a powerful tool. This can be done in MATLAB, as carried out in this section. Complete MATLAB script for factorial design and generating interaction plot can be seen in Annex A.3.

The plots show mean values. As it can be seen in Figure 76, the mean values of ultimate loads for different diameters and bolt spacing were given on the vertical axis, while input variables of the factor was given on the horizontal axis. For example, plot line with blue color represent the factor interaction of model with bolt spacing *b*=3, for diameter *d*=500, 700, 900. Whereas, the green color and red color plots represent models with bolt spacing *b*=4 and *b*=5, respectively. Taking one point on the plot, e.g. bolt spacing *b*=3 and diameter *d*=900, it will give a mean value of ultimate load of models 1-4-3-1-3, 1-4-3-2-3, 1-4-3-3-3, 1-4-5-1-3, 1-4-5-2-3, and 1-4-5-3-3, which is 12.6x10^6*N*.

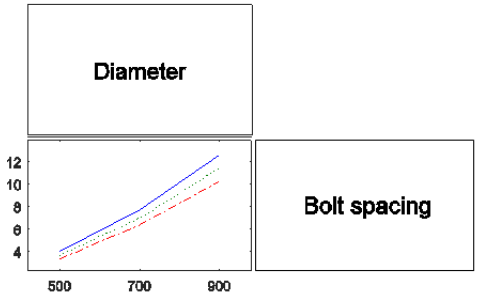


Figure 76. Main effects and interaction plot

This way the important factors and interaction between factors can be identified. Main effects may be identified directly from the response variable attributed to the factor, while the interaction exists if the plot lines are not parallel. A 2^k design (full factorial) will always give more information than one-factor-at-a-time (OFAT) method, where interaction can never be identified. If the OFAT method is used, the measured response will always confound main interaction effects (if interaction exist), which may lead to wrong conclusion. The results from OFAT method are only valid for those fixed values of the other factors.

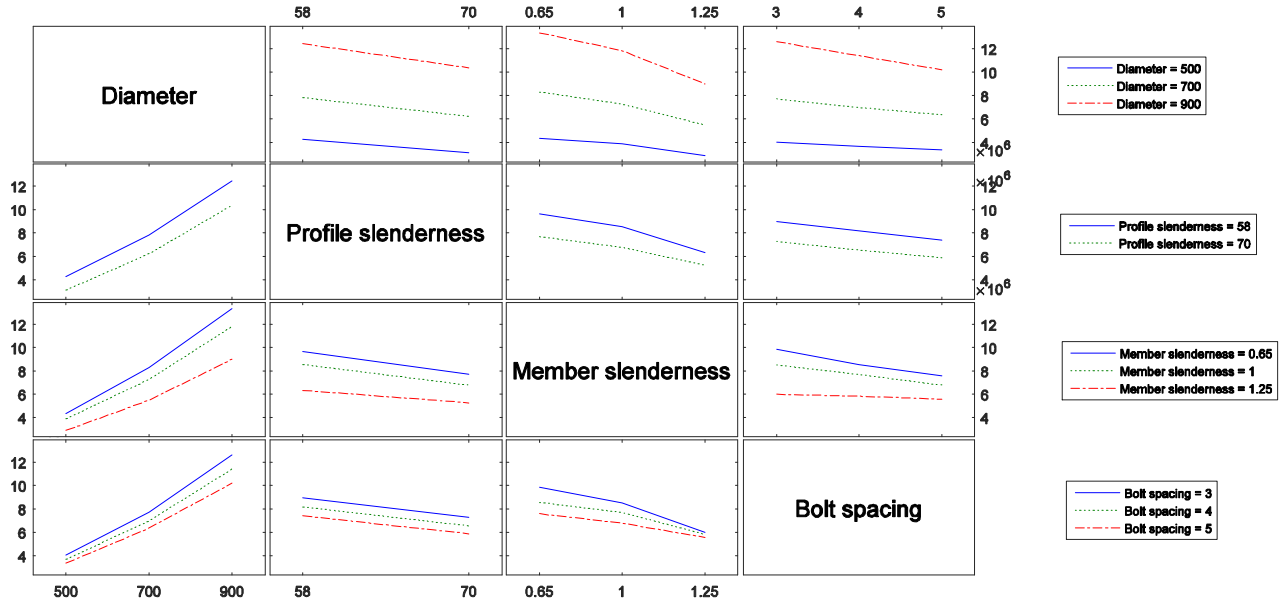


Figure 77. Interaction plot of factorial design for Riks-N Models

Figure 77 shows the interaction plot of full-factorial design with four factors for RIKS-N models. From the plot it can be concluded that diameter (*d*-parameter) have the most significant influence on the ultimate strength, and followed by member slenderness (λ-parameter). Profile slenderness (*slend*-parameter) and bolt spacing (*b*-parameter)have relatively moderate influence. No significant interaction exists, however interaction diameter-member slenderness and interaction member slenderness-bolt spacing are the highest among all interaction. Interaction between profile slenderness with other factors seemed very small which means profile slenderness dependency is negligible. Visible effect of interaction are especially shown when the factors interact with member slenderness $\bar{\lambda}$=1.25. It might suggest that the range used in the factors were not large enough.

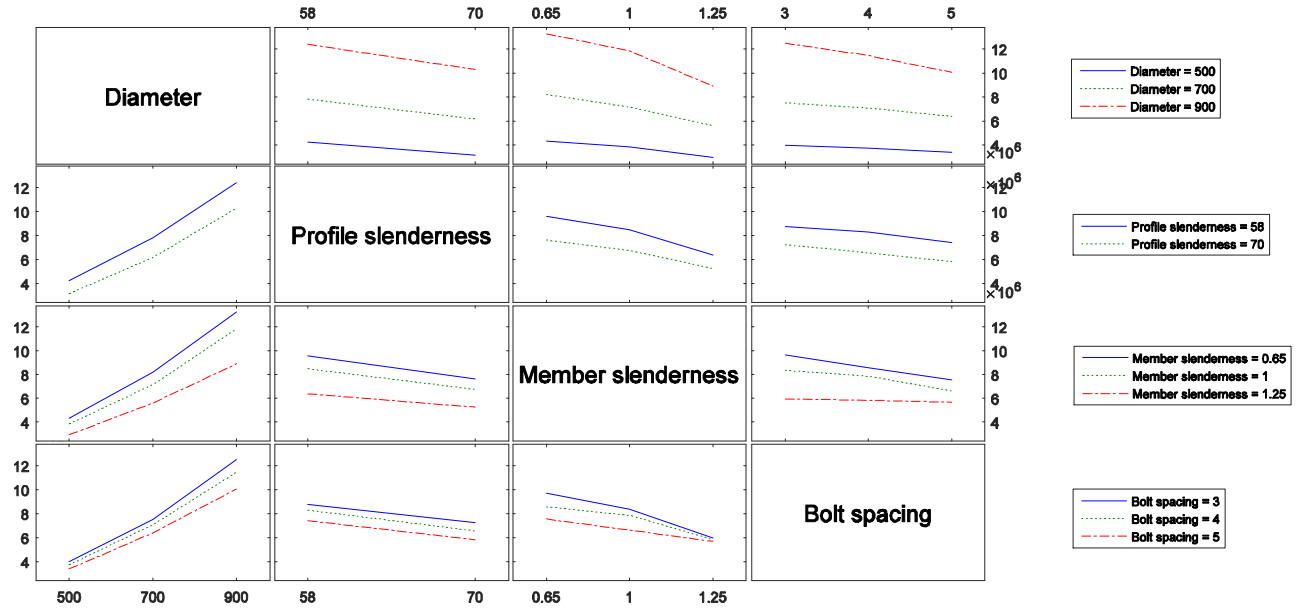


Figure 78.Interaction plot of factorial design for Riks-NM10 Models

Figure 78 shows the interaction plot for models subject to combined axial compression force and bending moment. It can be seen that the plot shows the same trend as for models under pure axial compression. The results may conclude that with application of relatively small bending moment, e.g. $0.1M_u$, no significant change on the ultimate strength expected to occur. The magnitude of applied bending moment might be too small for a significant influence, however ideally no bending moment should exist in the truss member. Small bending moment was applied in order to take into account moment at the joint due to unbalance force from diagonals, which may be found in practical condition.

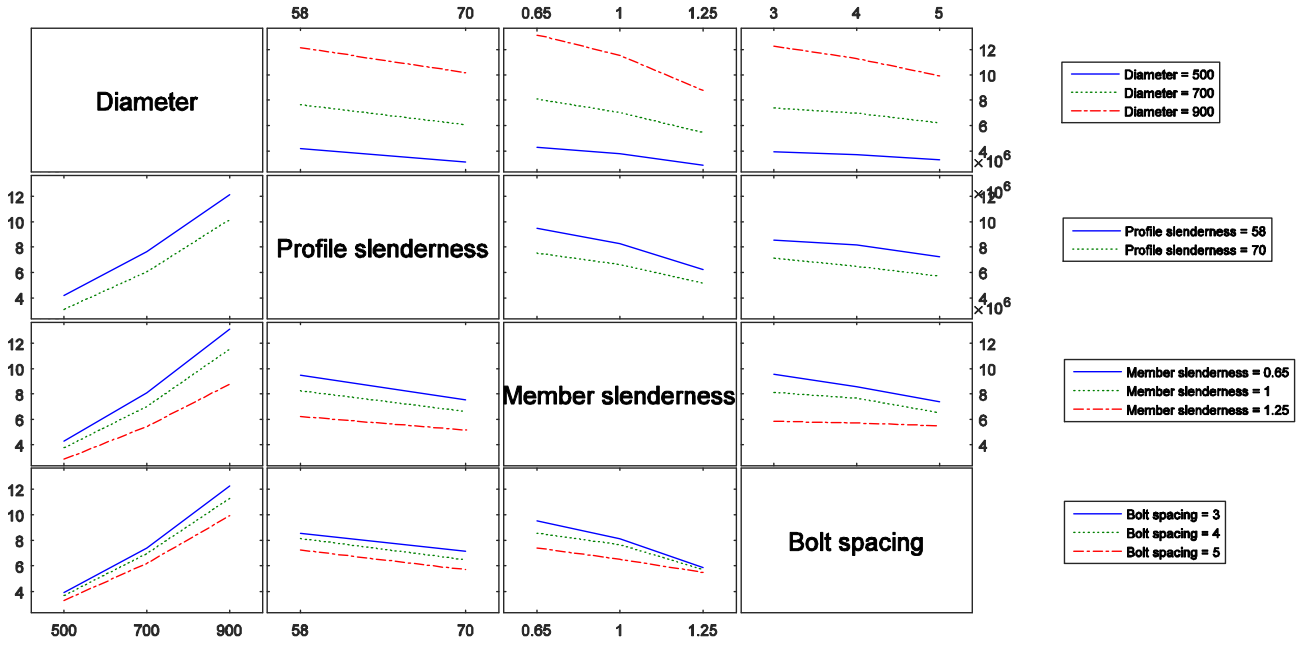


Figure 79. Interaction plot of factorial design for Riks-NM15 Models

Figure 79 and 80 shows the interaction plot for models RIKS-NM15 and RIKS-NM-05, respectively. No significant difference between them and the models with pure axial compression, i.e. RIKS-N.

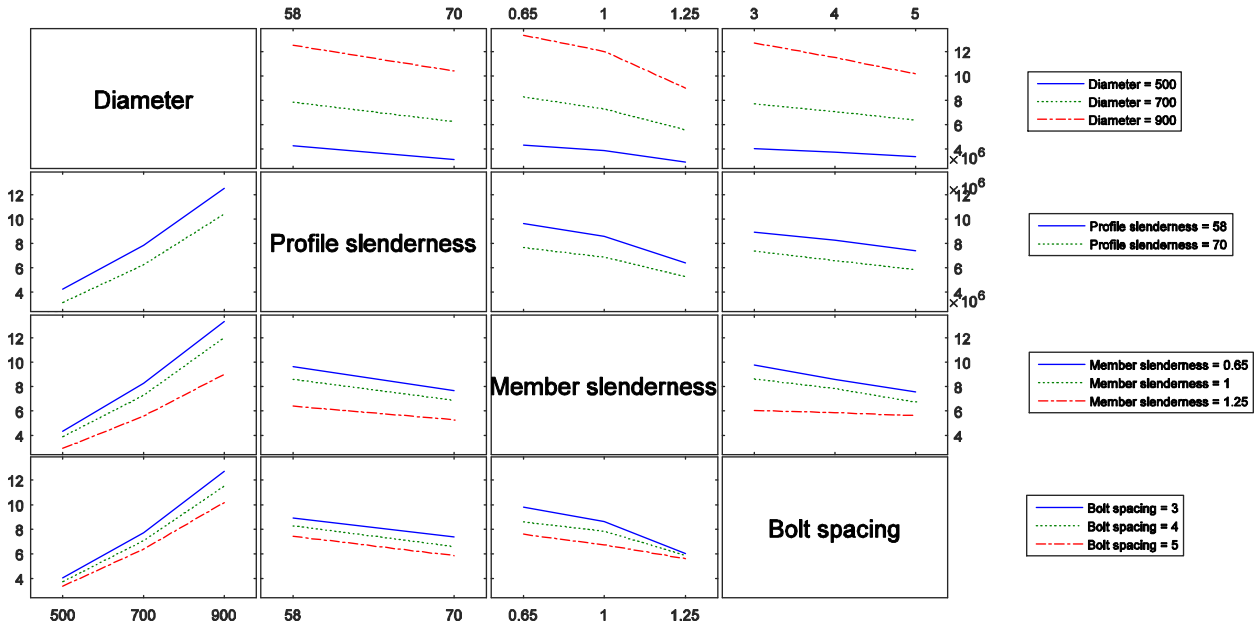


Figure 80. Interaction plot of factorial design for Riks-NM05 Models

4.3 FE Elastic Buckling and Non-linear Analysis

4.3.1 Verification of Elastic Buckling Analysis

In order to analyse the behaviour of the built-up columns the methodology as for the whole chord column (elastic buckling and non-linear post-buckling analysis, see chapter 2.3 and 2.7) were used. The analysis of elastic buckling was intended as the initial step of analysis for predicting the ultimate resistance of the proposed members. Since geometrical non-linearity, non-linear material, and imperfection exist and play an important influence on the ultimate resistance of this type of members, elastic buckling analysis cannot give accurate prediction of the resistance of the member even though it is related to the resistance. Therefore, a geometrical and material non-linear analysis with imperfection (GMNIA) was used for this purpose. However, determination of an accurate elastic buckling load and mode shape is important to the existing design method. The correlation between the elastic buckling and ultimate resistance of cold-formed members provides the basis for the design strength, e.g. when using Direct Strength Method. In this study, elastic buckling analysis was used, among others, to give imperfection mode shape for the non-linear analysis and calculation of member slenderness.

In the non-linear analysis, imperfections based on the mode shape from the buckling analysis have been introduced in accordance to EC 1993-1-5 [**10**]. For these calculations a value of s/2000 (*s* is the spacing of lip's bolt connection) was taken for the first four mode shapes from elastic buckling analysis, and then was used as initial imperfection. Distance *s* was chosen since sectorial buckling was the expected critical buckling mode in these models. Details of modelling the imperfection are described in Chapter 3.4.

Annex C.1 shows the first buckling mode shapes of all 54 models for the considered parameters. Example of buckling mode shape is shown in Figure 81. Buckling mode shape of the models with the lowest non-dimensional global slenderness ($\bar{\lambda}$), i.e. 0.65 are characterized by distortional buckling of the sector plate in between the lip's bolt connection, whereas higher non-dimensional slenderness, i.e. 1.0 and 1.25 were dominated by flexural buckling and interaction flexural-distortional buckling. This phenomenon occurs since $\bar{\lambda}$ correlated with length of the member or half-wave length of the buckling. Moreover, no torsional or torsional-flexural mode govern as the lowest critical buckling load which indicate that in these column models the torsional stiffness is almost fully developed. This semi-closed configuration of section member provides effective way to drastically increase the torsional rigidity, compared to the one as individual plate.

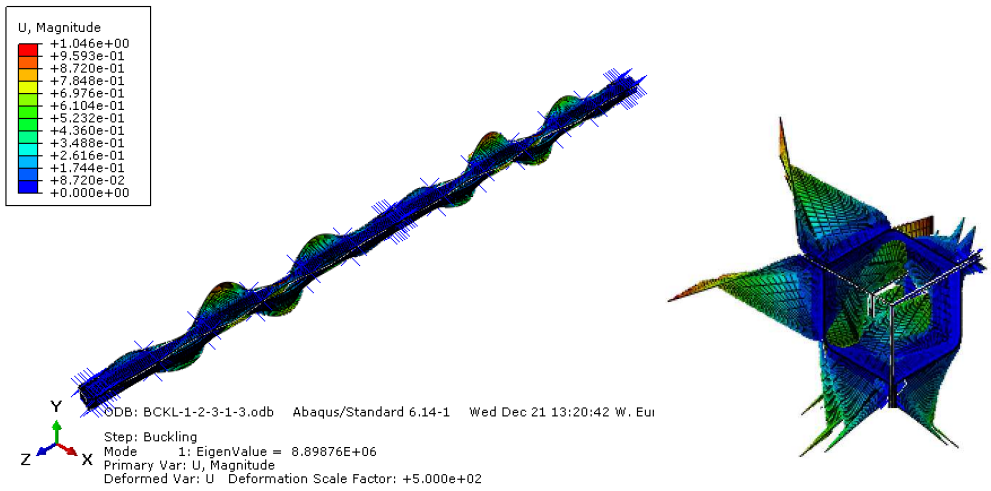


Figure 81. First buckling shape of Model 1-2-3-1-3

Results of elastic buckling analysis for all models are presented in Table 9. It can be seen from the table that the theoretical or Euler elastic buckling calculations based on EC1993-1-1 have a disagreement on the FEM analysis critical load for models with distortional-controlled buckling. In this case, the Euler critical loads are much higher than results from FEM analysis since the Euler buckling formulae consider the member as a whole perfect column and for flexural case only, without taking into account local or distortional buckling and interaction between them. In other hand, FEM analysed the member as a built-up plates connected together as semi-closed section, considering the real geometrical conditions. Therefore, the effect of non-fully rigid connected plates to the buckling behaviour was taken into account in FEM. With increasing bolt spacing, the critical loads decrease, and consequently the ratio of FEM-to-Euler critical load increases.

In case of models with flexural-controlled buckling, results from FEM are slightly higher that the theoretical one. This happens since effect of gusset plates in FEM model on the two end of the member will a bit increase the inertia of the member. Meanwhile, for two models, i.e. 1-4-3-2-4 and 1-4-5-2-4 the critical loads are slightly lower than the theoretical one. This happens due to interaction with distortional buckling (see picture in Annex E). Therefore, it is also noted that effect of the presence of another buckling type, e.g. distortional or local buckling, on the interaction of dependent buckling causes the critical load to be lower than the independent (individual) buckling.

In FEM, buckling analyses were performed for four eigenvalues and mode shapes for all models. It was shown that from the other buckling mode shapes, the deformation of the models were changing and unsymmetrical. It proves that in some cases, choosing a suitable buckling mode shape is complicated. It will influence on the cost of performing the analysis and accuracy of the results.

Therefore, in this study the imperfection took the first until the fourth mode shapes in order to make it more realistic.

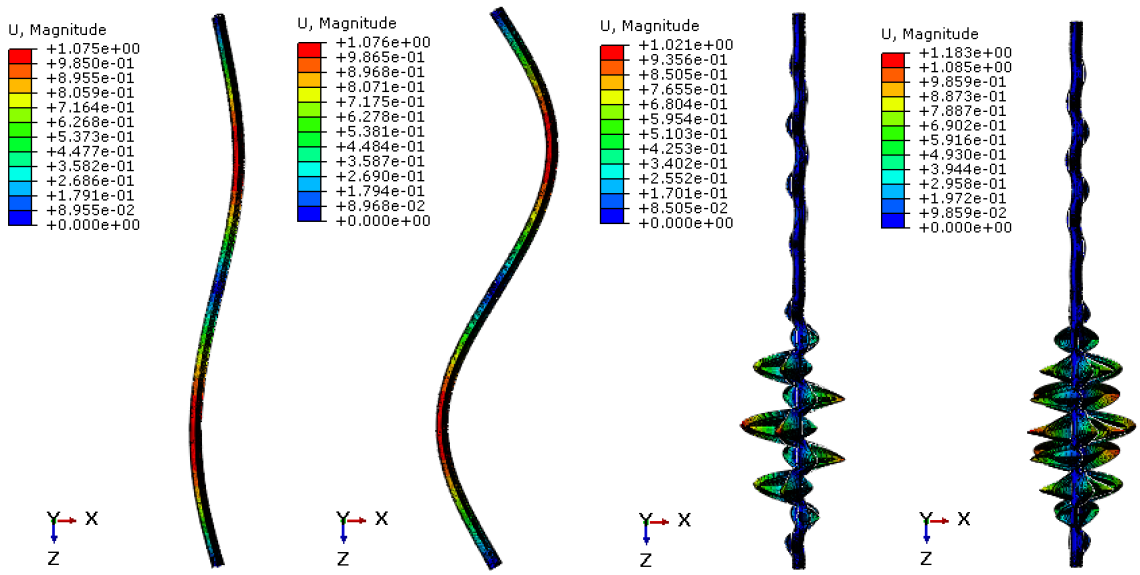


Figure 82. First four buckling mode shape of Model 1-2-3-2-3

Table 9.1st critical load according to elastic buckling analysis

(a) d-parameter=500

Model ID	Dia.	Thick.	Slend.	Length	bolt spacing	Elastic theory (Euler buckling)	Numerical Analysis (FEM)	Buckling mode
	d [mm]	t [mm]	λ [-]	l [mm]	s/d [-]	N_{crit} [kN]	N_{crit} [kN]	
1 2 3 1 3	500	9	0.65	18098	3	11095	8899	dist.
1 2 3 1 4					4	11095	6645	dist.
1 2 3 1 5					5	11095	5357	dist.
1 2 3 2 3			1	27035	3	4972	5099	flex.
1 2 3 2 4					4	4972	5021	flex.
1 2 3 2 5					5	4972	4889	dist.
1 2 3 3 3			1.25	33419	3	3254	3382	flex.
1 2 3 3 4					4	3254	3350	flex.
1 2 3 3 5					5	3254	3308	flex.
1 2 5 1 3		7	0.65	18106	3	8692	5995	dist.
1 2 5 1 4					4	8692	4413	dist.
1 2 5 1 5					5	8692	3540	dist.
1 2 5 2 3			1	27048	3	3895	4001	flex.
1 2 5 2 4					4	3895	3936	flex.
1 2 5 2 5					5	3895	3182	dist.
1 2 5 3 3			1.25	33435	3	2549	2654	flex.
1 2 5 3 4					4	2549	2628	flex.
1 2 5 3 5					5	2549	2594	flex.

(b) d-parameter=700

Model ID	Dia.	Thick.	Slend.	Length	bolt spacing	Elastic theory (Euler buckling)	Numerical Analysis (FEM)	Buckling mode
	d [mm]	t [mm]	λ [-]	l [mm]	s/d [-]	N_{crit} [kN]	N_{crit} [kN]	
1 3 3 1 3	700	12	0.65	25339	3	20741	15768	dist.
1 3 3 1 4					4	20741	11605	dist.
1 3 3 1 5					5	20741	9277	dist.
1 3 3 2 3			1	37852	3	9295	9487	flex.
1 3 3 2 4					4	9295	9333	flex.
1 3 3 2 5					5	9295	8383	dist.
1 3 3 3 3			1.25	46790	3	6083	6300	flex.
1 3 3 3 4					4	6083	6239	flex.
1 3 3 3 5					5	6083	6157	flex.
1 3 5 1 3		10	0.65	25347	3	17374	11900	dist.
1 3 5 1 4					4	17374	8683	dist.
1 3 5 1 5					5	17374	6907	dist.
1 3 5 2 3			1	37865	3	7785	7933	flex.
1 3 5 2 4					4	7785	7797	flex.
1 3 5 2 5					5	7785	6169	dist.
1 3 5 3 3			1.25	46806	3	5095	5271	flex.
1 3 5 3 4					4	5095	5218	flex.
1 3 5 3 5					5	5095	5147	flex.

(c) d-parameter=900

Model ID	Dia.	Thick.	Slend.	Length	bolt spacing	Elastic theory (Euler buckling)	Numerical Analysis (FEM)	Buckling mode
	d [mm]	t [mm]	λ [-]	l [mm]	s/d [-]	N_{crit} [kN]	N_{crit} [kN]	
1 4 3 1 3	900	15	0.65	32580	3	33361	24696	dist.
1 4 3 1 4					4	33361	17943	dist.
1 4 3 1 5					5	33361	14242	dist.
1 4 3 2 3			1	48669	3	14950	15182	flex.
1 4 3 2 4					4	14950	14924	flex-dist
1 4 3 2 5					5	14950	12777	dist.
1 4 3 3 3			1.25	60161	3	9784	10095	flex.
1 4 3 3 4					4	9784	9992	flex.
1 4 3 3 5					5	9784	9857	flex.
1 4 5 1 3		13	0.65	32588	3	29029	19850	dist.
1 4 5 1 4					4	29029	14355	dist.
1 4 5 1 5					5	29029	13347	dist.
1 4 5 2 3			1	48682	3	13008	13193	flex.
1 4 5 2 4					4	13008	12956	flex-dist
1 4 5 2 5					5	13008	10076	dist.
1 4 5 3 3			1.25	60177	3	8513	8777	flex.
1 4 5 3 4					4	8513	8685	flex.
1 4 5 3 5					5	8513	8563	flex.

The comparison of elastic critical load between the models with predefined parameters can be seen in Table 9. Critical load of the models are significantly higher for larger diameter as the inertia increases. With the same cross sectional slenderness, the average increases of critical load are 90.8% and 62.8% for changing diameter 500*mm* to 700*mm* and diameter 700*mm* to 900*mm*, respectively. The thicknesses of plates also affect the critical load. An average increase of 17.6% for models d=900*mm* with thickness from 13*mm* to 15*mm*. Higher slenderness, consequently longer member, has lower critical load. An average decrease of 36.9% on critical load was experienced for models by changing$\bar{\lambda}$=0.65 into $\bar{\lambda}$=1.0, and 33.6% decrease of critical load was experienced for models by changing $\bar{\lambda}$=1.0 into $\bar{\lambda}$=1.25. Effect of bolt spacing on the critical load depends on the length of member. The effect is significant for short member, in this case $\bar{\lambda}$=0.65, with average decrease of 26.7% and 17.9% by changing the bolt spacing from *3d* to *4d* and *4d* to *5d*, respectively. For member with $\bar{\lambda}$=1.0 an average decrease of 1.7% for changing bolt spacing from *3d* to *4d*, meanwhile for member with $\bar{\lambda}$=1.25 an average decrease of 0.9% for changing bolt spacing from *3d* to *4d*.

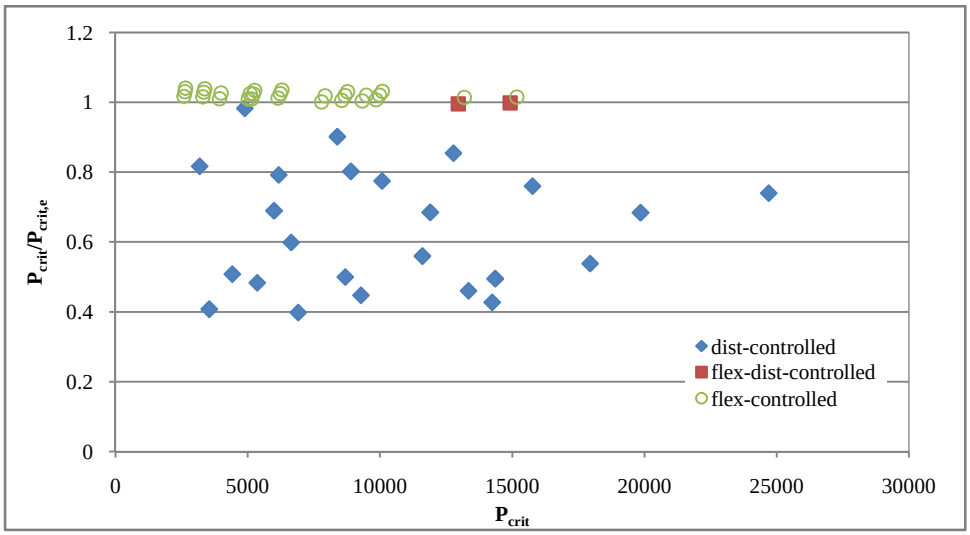


Figure 83. P_{crit}-to-P_e ratio

This analysis shows that the bolt spacing of *3d – 5d* on this type of polygonal sections gives elastic buckling modes of distortional, flexural, and flexural-distortional. With this bolt density, influence of the stiffness of the lip connection on triggering different elastic buckling type, i.e. local buckling is not sufficient since result shows that all buckling modes lies in range of distortional and flexural type.

Elastic buckling prediction by current design specification (i.e. EC1993) for local and flexural mode is only appropriate when the members are wholly thin plate and column, respectively. Moreover, for distortional buckling prediction there is no explicit expression available. In this case, where the members are semi-closed built-up column composed of folded plates, there is no expressions in Eurocode for predicting the elastic critical buckling, either for sectorial or global buckling modes. An element model and semi-empirical model for elastic buckling prediction of open cross-sections was developed by AISI (1996), Lau and Hancock (1987), and Schafer (2008). Application of the latter calculation method can be found in finite-strip method, e.g. CUFSM. Therefore, models and expressions of elastic buckling for semi-closed cross-sections need to be developed and evaluated to determine whether a similar correlation to that for open cross-section also exists for semi-closed cross-section.

Using elastic buckling loads obtained from FE models as the basis for semi-closed polygonal columns would be ineffective, because it would have to exert enormous effort to create an FE model and then inspect and choose the appropriate buckling mode. Limited number of models run in this parametric study made it not feasible to provide equations for the elastic buckling loads.

4.3.2 Verification of Non-linear Analysis

4.3.2.1 General

The non-linear finite element analysis using Riks solver was carried out for analyses of unstable, materially and geometrically nonlinear with imperfections considered. The ultimate load was evaluated and compared between the models. ABAQUS models uses four-node general-purpose shell element with finite membrane strains, elastic-plastic material with strain hardening, and imperfections as suggested in the modeling guidelines for cold-formed steel [**17**] were created.

A total of 216 non-linear collapse analyses were carried out for all column models with predefined parameters and under pure axial compression (N) and combined axial-bending moment (NM0.05, NM0.1, and NM0.15), of which all analyses successfully reached the ultimate loads. Table 10shows the ultimate load and corresponding displacement for all models which includes the defining variables and types of failure. The results were obtained by Python automation from ABAQUS as described in Chapter 4.1.

Table 10. Ultimate loads and corresponding displacement from FE non-linear analysis for models RIKS-N

(a) d-parameter=500

Model ID	Dia.	Thick.	Slend.	Length	bolt spacing	Max. load	Shortening	Normalized resistance	Failure mode
	[mm]	[mm]	[-]	[mm]	[-]	P_u [kN]	disp [mm]	P_u/P_{yg} [-]	
1 2 3 1 3	500	9	0.65	18098	3	5625.7	29.13	1.00	dist.
1 2 3 1 4					4	4982.5	26.03	0.89	dist.
1 2 3 1 5					5	4531.8	24.75	0.81	dist.
1 2 3 2 3			1	27035	3	4865.7	37.93	0.87	dist.
1 2 3 2 4					4	4555.0	35.44	0.82	dist.
1 2 3 2 5					5	4126.2	32.17	0.74	dist.
1 2 3 3 3			1.25	33419	3	3335.7	34.09	0.60	dist-flex
1 2 3 3 4					4	3269.5	33.10	0.59	dist-flex
1 2 3 3 5					5	3176.2	31.49	0.57	dist-flex
1 2 5 1 3		7	0.65	18106	3	4132.5	27.34	0.95	dist.
1 2 5 1 4					4	3576.7	24.21	0.82	dist.
1 2 5 1 5					5	3161.9	23.20	0.72	dist.
1 2 5 2 3			1	27048	3	3665.2	36.35	0.84	dist.
1 2 5 2 4					4	3241.9	32.20	0.74	dist.
1 2 5 2 5					5	2840.9	28.59	0.65	dist.
1 2 5 3 3			1.25	33435	3	2606.7	33.78	0.60	dist-flex
1 2 5 3 4					4	2496.8	31.14	0.57	dist-flex
1 2 5 3 5					5	2405.3	30.03	0.55	dist-flex

(b) d-parameter=700

Model ID	Dia.	Thick.	Slend.	Length	bolt spacing	Max. load	Shortening	Normalized resistance	Failure mode
	[mm]	[mm]	[-]	[mm]	[-]	P_u [kN]	disp [mm]	P_u/P_{yg} [-]	
1 3 3 1 3	700	12	0.65	25339	3	10583.1	41.10	1.00	dist.
1 3 3 1 4					4	9053.7	35.59	0.87	dist.
1 3 3 1 5					5	8140.8	33.86	0.78	dist.
1 3 3 2 3			1	37852	3	8974.8	52.52	0.86	dist.
1 3 3 2 4					4	8128.1	47.34	0.78	dist.
1 3 3 2 5					5	7400.3	43.45	0.71	dist.
1 3 3 3 3			1.25	46790	3	6192.9	47.06	0.59	dist-flex
1 3 3 3 4					4	6020.5	45.09	0.58	dist-flex
1 3 3 3 5					5	5901.9	43.72	0.57	dist-flex
1 3 5 1 3		10	0.65	25347	3	8309.6	38.52	0.95	dist.
1 3 5 1 4					4	7358.9	34.13	0.84	dist.
1 3 5 1 5					5	6417.5	32.86	0.74	dist.
1 3 5 2 3			1	37865	3	7105.3	49.55	0.81	dist.
1 3 5 2 4					4	6373.9	44.55	0.73	dist.
1 3 5 2 5					5	5661.1	40.01	0.65	dist.
1 3 5 3 3			1.25	46806	3	5177.7	47.83	0.59	dist-flex
1 3 5 3 4					4	4985.4	44.02	0.57	dist-flex
1 3 5 3 5					5	4683.0	40.90	0.54	dist-flex

(c) d-parameter=900

Model ID	Dia.	Thick.	Slend.	Length	bolt spacing	Max. load	Shortening	Normalized resistance	Failure mode
	[mm]	[mm]	[-]	[mm]	[-]	P_u [kN]	disp [mm]	P_u/P_{yg} [-]	
1 4 3 1 3	900	15	0.65	32580	3	16670.0	51.85	0.99	dist.
1 4 3 1 4					4	14571.6	45.18	0.87	dist.
1 4 3 1 5					5	12630.0	41.79	0.75	dist.
1 4 3 2 3			1	48669	3	14330.1	66.74	0.86	dist.
1 4 3 2 4					4	13158.7	61.34	0.79	dist.
1 4 3 2 5					5	11412.8	53.88	0.68	dist.
1 4 3 3 3			1.25	60161	3	9958.4	60.53	0.59	dist-flex
1 4 3 3 4					4	9777.5	58.85	0.58	dist-flex
1 4 3 3 5					5	9320.6	54.80	0.56	dist-flex
1 4 5 1 3		13	0.65	32588	3	13828.6	49.28	0.95	dist.
1 4 5 1 4					4	11788.0	43.61	0.81	dist.
1 4 5 1 5					5	10664.4	42.53	0.73	dist.
1 4 5 2 3			1	48682	3	12124.6	65.03	0.83	dist.
1 4 5 2 4					4	10701.6	57.72	0.73	dist.
1 4 5 2 5					5	9216.9	50.32	0.63	dist.
1 4 5 3 3			1.25	60177	3	8656.9	61.02	0.59	dist-flex
1 4 5 3 4					4	8400.3	57.10	0.58	dist-flex
1 4 5 3 5					5	7913.5	53.21	0.54	dist-flex

From the above tables, it can be seen that the failure mode of the models are dominated by distortional failure with the remaining models show interaction distortional-flexural failure. Annex C.2 provides the screenshots of the models at failure state taken from ABAQUS non-linear analysis. The difference of this result to the elastic buckling analysis can be clearly noticed, where there is no independent global flexural failure mode in non-linear analysis. This affirms the significant influence of material and geometrical non-linearity and imperfections on the ultimate resistance of the studied models. The results for combined axial compression- bending moment (RIKS-NM) models are presented in the next section.

The results of the parametric studies of all column models are presented in Figure 84 in the form of normalized resistance ($P_{u\text{-}FEM}/P_{yg}$) based on fully effective cross-section resistance (P_{yg}) versus local slenderness $\lambda_{cr\text{-}FEM}=(P_{yg}/P_{cr\text{-}FEM})^{0.5}$ based on P_{yg} and the critical buckling load by the FEM analysis. It is important to note that the failure mode of all models in this non-linear study is distortional type, while some of the first critical buckling loads by FEM discussed in previous section (4.3.1) were the flexural type. Therefore, it is needed to obtain distortional buckling loads for those models in order to have proper local slenderness, according to the actual failure mode. This is performed by looking at other buckling modes which give distortional type of buckling in ABAQUS. Then, these critical buckling loads (Table 11) were taken for constructing the graph. Figure 84 also shows the EC1993-1-3 [**9**] resistance curve for distortional buckling mode, and other codes for different buckling modes.

From the graph it is worth noted that there are two obvious scatters of data points when slenderness $0.9<\lambda_{cr\text{-}FEM}<1.2$. The higher scatter group corresponds to those columns which failed in a pure distortional mode, while the lower scatter corresponds to those columns failed in distortional-flexural interaction. This means although the vast majority of the failure modes from the FEM are distortional modes, an interaction distortional-flexural failure did occur for models with high global slenderness, $\bar{\lambda}$ = 1.25, in which give lower strength than the distortional modes. Figure 84 shows that in case of distortional failure, a full proportion of the numerical ultimate resistance were slightly underestimated by the Code, which means that EC1993-1-3 corresponds to the FEM analysis results and gives safe but less conservative prediction. It can be said that the EC1993-1-3 curve almost forms the lower bound of the numerical ultimate resistance for the studied models. The detail discussion and verification of analysis based on design code specification (EC1993) is given in the next section.

A small scatters of data points which appeared below the EC1993-1-3 strength curve in the graph are all models with global slenderness $\bar{\lambda}$ = 1.25, and a safe estimates of strength cannot be provided by EC1993-1-3 since it is an interaction modes. Very unsafe predictions were provided by the EC1993-1-3 curve for those models. These data points fell on local slenderness, i.e. $\lambda_{cr\text{-}FEM}\geq 0.9$ for models with bolt spacing ratio, b = 4 and larger slenderness $\lambda_{cr\text{-}FEM}\geq 1.0$ for models with b = 5. The flexural slenderness which was significantly high and larger than the distortional slenderness may induce the flexural deformation interacted with the distortional mode. It is suggested by this parametric study that the EC1993-1-3 curve provides safe (or almost safe) predictions if the columns fail in a pure distortional mode, whereas give unsafe prediction if the columns fail in interaction mode, D-F interaction in this case.

By travelling from the highest to the lowest $P_{u\text{-}FEM}/P_{yg}$ values, the scatter of data points correspond to the increase of bolt spacing ratio *b*-parameter. This shows that with the increase of bolt spacing, being from b=*3* to *b*=4 and *b*=5, the ultimate resistance tends to decrease.

This shows that the non-dimensional slenderness and bolt spacing are the most influencing parameter which characteristics resulted in significantly deviated ultimate resistance. Likewise, the diameter *d*-

parameter and thickness *t*-parameter give unfavourable effect on the ultimate resistance when it increases, however this influence is not significant and considerably lower than those caused by slenderness and bolt spacing.

If Figure 84 is redrawn as Figure 85, which plots ($P_{u\text{-}FEM}/P_{yg}$) versus $\lambda_{cr\text{-}FEM}= (P_{yg}/P_{cr\text{-}FEM})^{0.5}$ for loading condition of combined axial compression-bending moment (RIKS-NM), the scatter of data points for the numerical ultimate resistance move slightly downward, which results in unsafe predictions for some models even though a large portion are still on the safe side. This happens as expected since the presence of bending moment will reduce the ultimate strength of the column due to load interaction. The axial-moment (NM) interaction of the studied columns will be discussed in detail in the next section.

From this analysis it can be noted that for the studied columns, distortional failures have lower post-buckling capacity than the other mode, i.e. local buckling. Furthermore, distortional buckling may control the failure mechanism even when the elastic distortional buckling stress (f_{crd}) is higher than the elastic flexural buckling stress (f_{crf}).

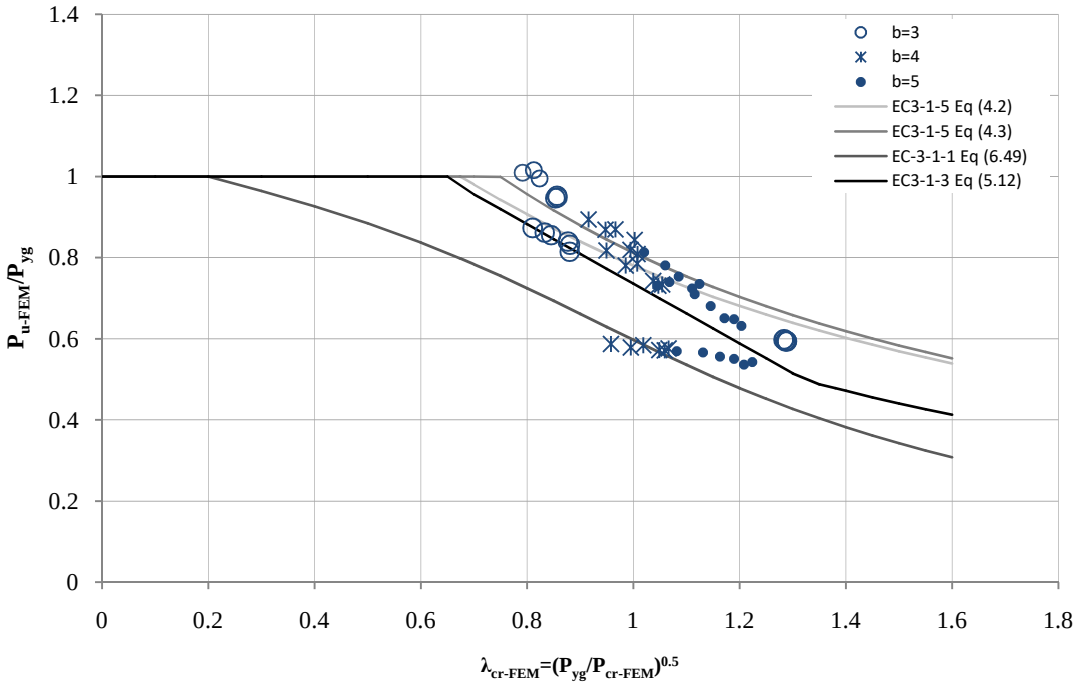


Figure 84. Parametric study results of the studied columns (normalized resistance based on P_{yg} vs. slenderness based on P_{yg} and FEM critical buckling loads): Models RIKS-N

As shown in the subchapter 4.3 Factorial design, the most significant factor interaction are member slenderness-bolt spacing and member slenderness-diameter, hence plot was also made for diameter group scatter, as shown below. As for diameter, the phenomenon of interaction distortional-flexural failure did occur for models with all diameters, i.e. 500, 700, and 900, when high global slenderness, $\bar{\lambda}$ = 1.25 possessed.

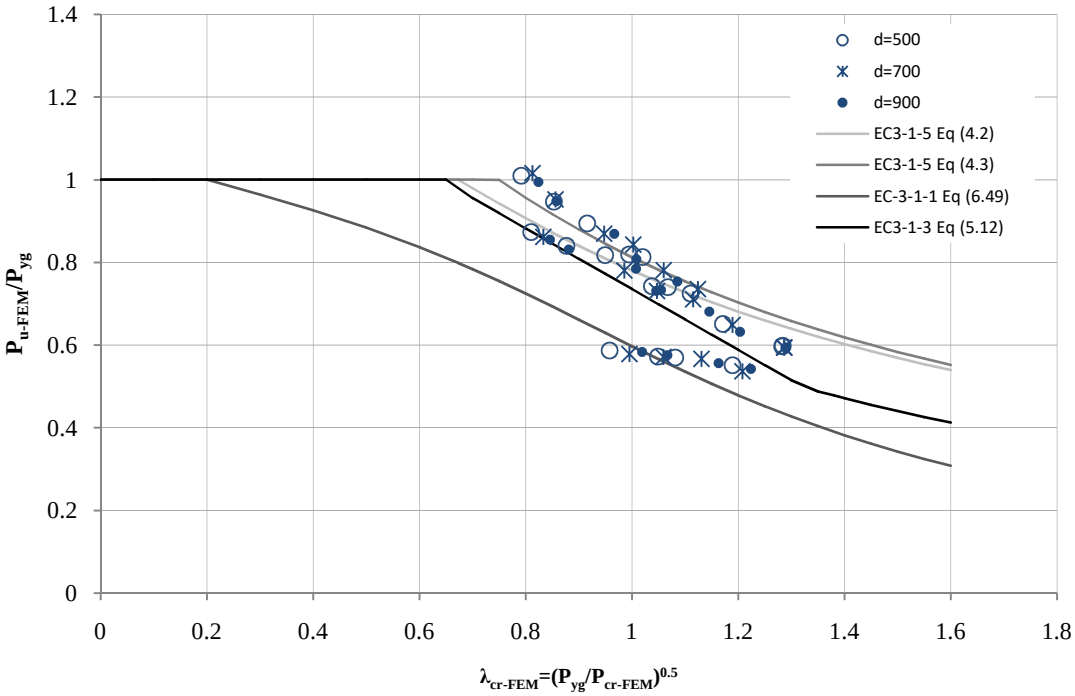


Figure 85. Parametric study results of the studied columns (normalized resistance based on P_{yg} vs. slenderness based on P_{yg} and FEM critical buckling loads): Models RIKS-N

As a sample for visualization of failure mode, Figure 86 shows a typical distortional buckling mode at the ultimate load of model RIKS-N-1-2-5-1-3. The colour contours represent the magnitudes of von Misses stress. Complete screenshots of failure mode for all models are given in Annex C.2.

The failure is characterized by buckling of the lips outward, while no significant rotation along its weak axis experienced by the entire column. This type of buckling is also known as “stiffener flexural buckling” or “local-torsional buckling”. Lips as connection between plates in this type of sections can be considered as the stiffener. Distortional mode well recognized by rotation of the flange at the flange-web junction or displacement of the intermediate stiffener normal to the plane of the element. The junction point between flange and web moves inward or outward, as exhibited in the result of numerical analysis of this parametric study. Distortional buckling exists at intermediate longitudinal half sine waves (half-wavelength), between short local buckling half-wavelength and long flexural or flexural-torsional buckling half-wavelength. In this case, the half-wavelength is the bolt spacing distance (*s*).

Meanwhile, Figure 87 shows the failure mode by interaction between distortional and flexural buckling for models with $\bar{\lambda}$ = 1.25. The failure mode is characterized by buckling of the lips outward and at the same time, large rotation of the entire member.

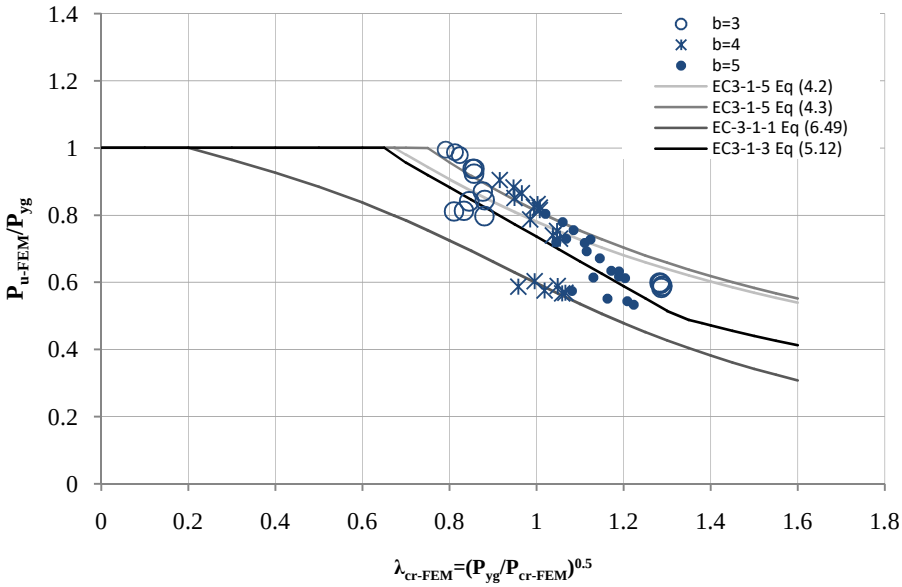
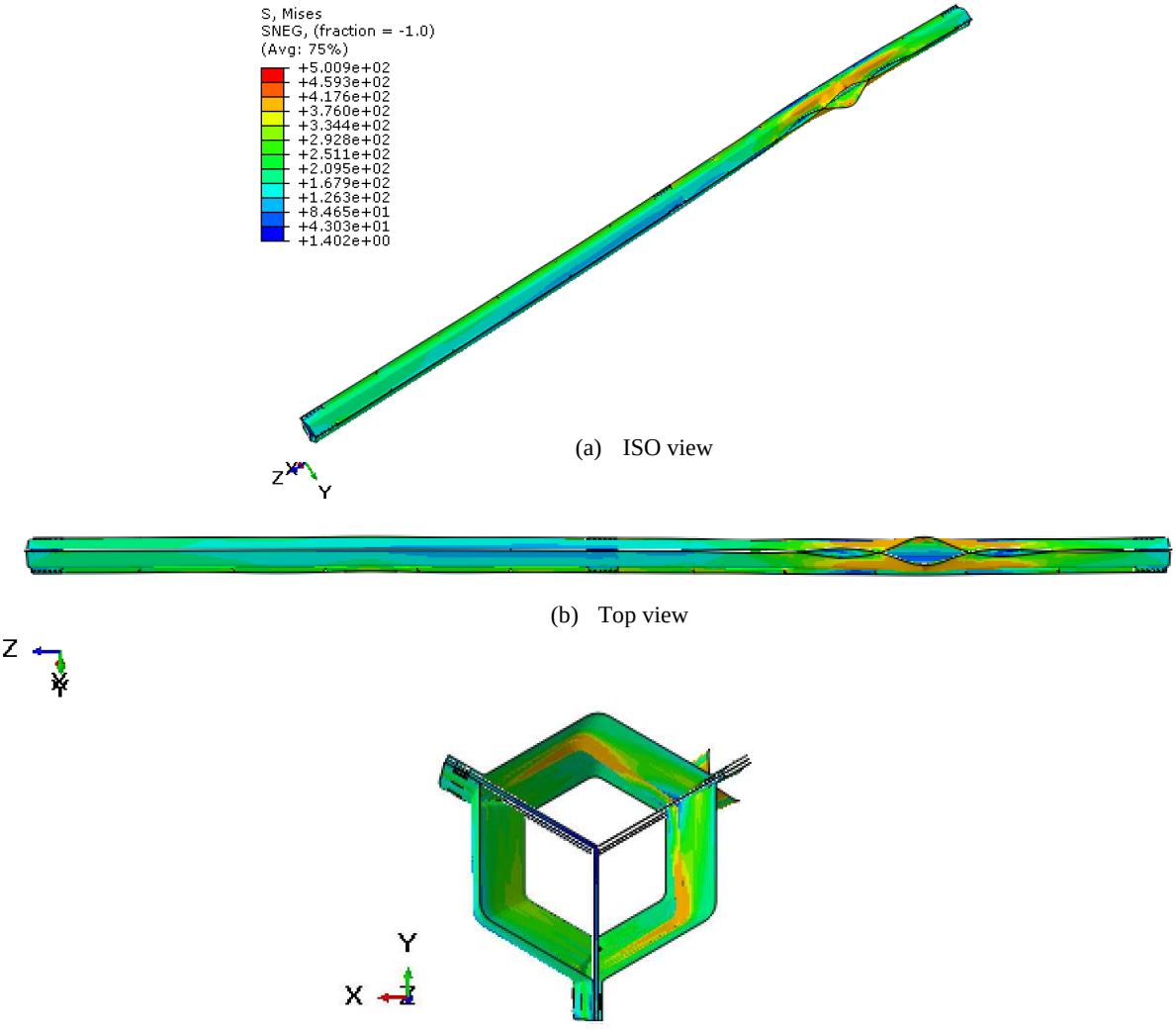


Figure 86. Parametric study results of the studied columns (normalized resistance based on P_{yg} vs. slenderness based on P_{yg} and FEM critical buckling loads): Models RIKS-NM01

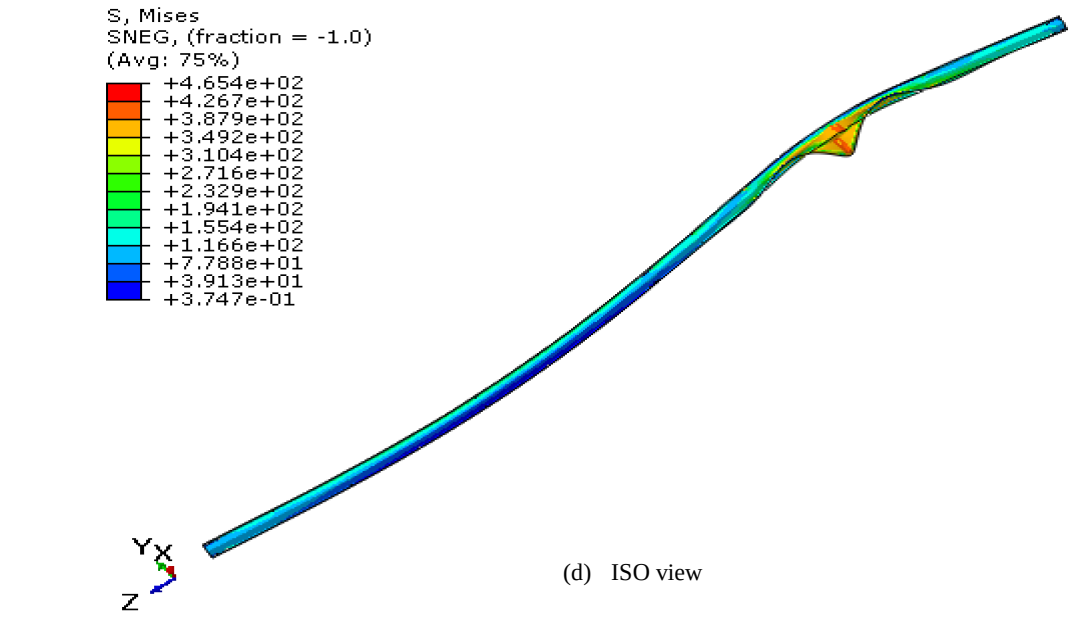


(a) ISO view

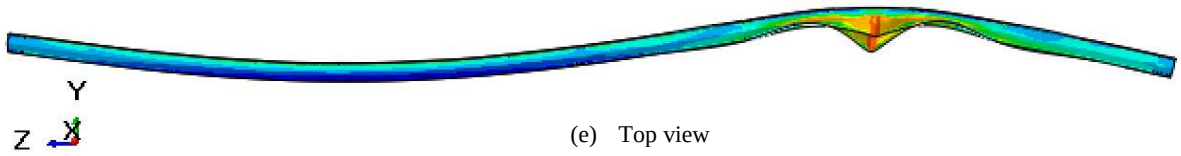
(b) Top view

(c) Front view

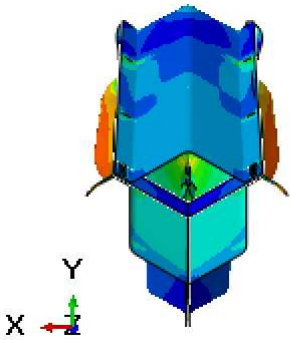
Figure 87. Failure mode of a RIKS-N model failing by distortional mode



(d) ISO view



(e) Top view



(f) Front view

Figure 88. Failure mode of a RIKS-N model failing by distortional-flexural interaction mode

Moreover, this parametric study is expected to rule out the limit of normalized resistance for the expected failure mode, in this case distortional buckling, based on slenderness of the member. From graph in Figure 81, it can be suggested that the expression used in EC1993-1-3 for reduction factor due to distortional buckling (Eq. 5.12) may be adopted for the semi-closed polygonal type of cross-section undergoes pure distortional buckling mode used in this parametric study, with provision of global slenderness, $\bar{\lambda}$< 1.25. The EC1993-1-3 became the lower bound for the results of numerical ultimate strength.

$$\chi_d = 1.0 \quad \text{if} \quad \bar{\lambda}_d<0.65; \ \ \bar{\lambda}< 1.25$$

$$\chi_d = 1.47 - 0.723\bar{\lambda}_d \quad \text{if} \quad 0.65<\bar{\lambda}_d<1.38; \ \ \bar{\lambda}< 1.25$$

$$\chi_d = \frac{0.66}{\bar{\lambda}_d} \quad \text{if} \quad \bar{\lambda}_d \geq 1.38; \ \ \bar{\lambda}< 1.25$$

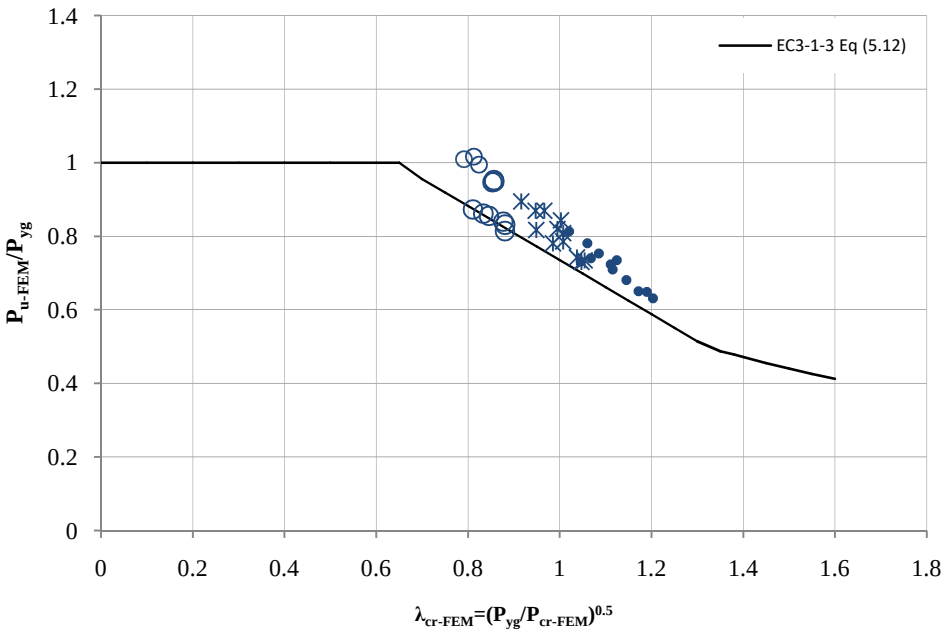


Figure 89. Parametric study results of the studied columns (normalized resistance based on P_{yg} vs. slenderness based on P_{yg} and FEM critical buckling loads): Pure distortional mode

Meanwhile, a distortional-flexural interaction would be found in models with global slenderness $\bar{\lambda}$ = 1.25. Therefore, by excluding the cluster data points of pure distortional mode, a linear regression was developed to get the expression of ultimate resistance:

$$\chi_{d-f} = 0.7576 - 0.1751\bar{\lambda}_d \quad \text{if} \quad 0.9<\bar{\lambda}_d<1.25;\ \bar{\lambda} \geq 1.25$$

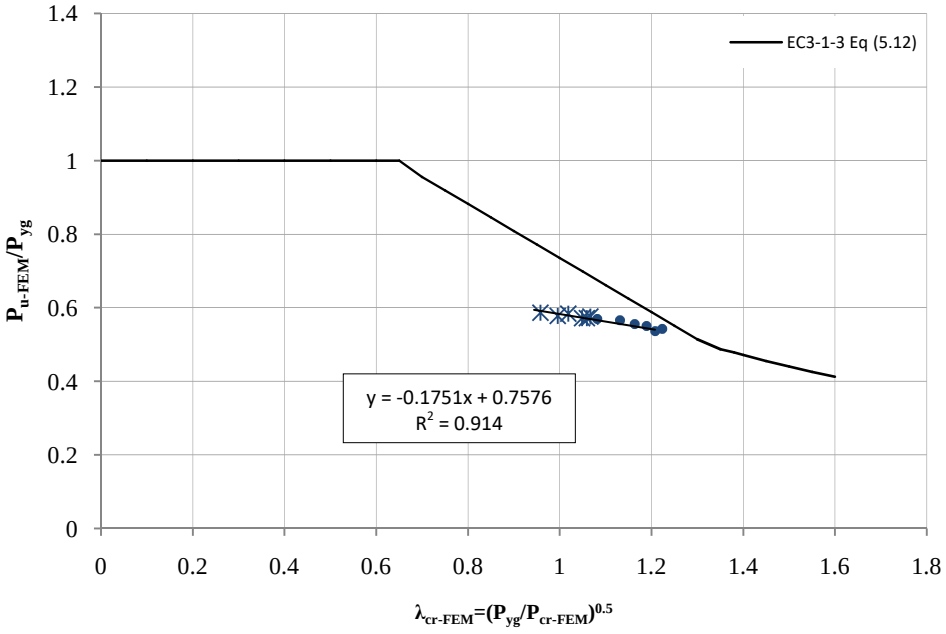


Figure 90. Parametric study results of the studied columns (normalized resistance based on P_{yg} vs. slenderness based on P_{yg} and FEM critical buckling loads): Distortional-flexural interaction

The critical buckling mode for various eigenmodes are shown in the following table.

Table 11. Critical buckling loads for various eigenmodes

(a) d-parameter=500

Model ID	Dia.	Thick.	Slend.	Length	bolt spacing	Numerical Analysis (FEM) Elastic buckling ($N_{crit.}$)		
	d [mm]	t [mm]	λ [-]	l [mm]	s/d [-]	Dist. [kN]	Flex-Dist [kN]	Flex. [kN]
1 2 3 1 3	500	9	0.65	18098	3	8899	-	-
1 2 3 1 4					4	6645	-	-
1 2 3 1 5					5	5357	-	-
1 2 3 2 3			1	27035	3	-	-	5099
1 2 3 2 4					4	-	-	5021
1 2 3 2 5					5	4889	-	-
1 2 3 3 3			1.25	33419	3	-	-	3382
1 2 3 3 4					4	-	-	3350
1 2 3 3 5					5	4769	-	3308
1 2 5 1 3		7	0.65	18106	3	5995	-	-
1 2 5 1 4					4	4413	-	-
1 2 5 1 5					5	3540	-	-
1 2 5 2 3			1	27048	3	-	-	4001
1 2 5 2 4					4	-	-	3936
1 2 5 2 5					5	3182	-	-
1 2 5 3 3			1.25	33435	3	-	-	2654
1 2 5 3 4					4	-	-	2628
1 2 5 3 5					5	-	-	2594

(b) d-parameter=700

Model ID	Dia.	Thick.	Slend.	Length	bolt spacing	Numerical Analysis (FEM) Elastic buckling ($N_{crit.}$)		
	d [mm]	t [mm]	λ [-]	l [mm]	s/d [-]	Dist. [kN]	Flex-Dist [kN]	Flex. [kN]
1 3 3 1 3	700	12	0.65	25339	3	15768	-	-
1 3 3 1 4					4	11605	-	-
1 3 3 1 5					5	9277	-	-
1 3 3 2 3			1	37852	3	-	-	9487
1 3 3 2 4					4	-	-	9333
1 3 3 2 5					5	8383	-	-
1 3 3 3 3			1.25	46790	3	-	-	6300
1 3 3 3 4					4	-	-	6239
1 3 3 3 5					5	-	-	6157
1 3 5 1 3		10	0.65	25347	3	11900	-	-
1 3 5 1 4					4	8683	-	-
1 3 5 1 5					5	6907	-	-
1 3 5 2 3			1	37865	3	-	-	7933
1 3 5 2 4					4	-	-	7797
1 3 5 2 5					5	6169	-	-
1 3 5 3 3			1.25	46806	3	-	-	5271
1 3 5 3 4					4	-	-	5218
1 3 5 3 5					5	-	-	5147

(c) d-parameter=900

Model ID	Dia.	Thick.	Slend.	Length	bolt spacing	Numerical Analysis (FEM) Elastic buckling ($N_{crit.}$)		
	d [mm]	t [mm]	λ [-]	l [mm]	s/d [-]	Dist. [kN]	Flex-Dist [kN]	Flex. [kN]
1 4 3 1 3	900	15	0.65	32580	3	24696	-	-
1 4 3 1 4					4	17943	-	-
1 4 3 1 5					5	14242	-	-
1 4 3 2 3			1	48669	3	-	-	15182
1 4 3 2 4					4	-	14924	-
1 4 3 2 5					5	12777	-	-
1 4 3 3 3			1.25	60161	3	-	-	10095
1 4 3 3 4					4	-	-	9992
1 4 3 3 5					5	-	-	9857
1 4 5 1 3		13	0.65	32588	3	19850	-	-
1 4 5 1 4					4	14355	-	-
1 4 5 1 5					5	13347	-	-
1 4 5 2 3			1	48682	3	-	-	13193
1 4 5 2 4					4	-	12956	-
1 4 5 2 5					5	10076	-	-
1 4 5 3 3			1.25	60177	3	-	-	8777
1 4 5 3 4					4	-	-	8685
1 4 5 3 5					5	-	-	8563

4.3.2.2 Load-Displacement of FE Analysis

Load-displacement curves based on unstable analysis of Riks method are presented in this section. The ultimate loads and resistance of the cold-formed sections according to EC-1993-1-3 are also determined. Note that due to space limitations, the results of the samples are provided here, while the complete data can be seen in Annex. Four predefined parameters, i.e. diameter (*d*), thickness (*t*), length ($\bar{\lambda}$), and bolt spacing (*b*) are compared to see the significance of them on the ultimate resistance of the members. Results from FEA and EC1993-1-3 ultimate loads are close as already observed in previous section (Figure 84).

4.3.2.2.1 Diameter (d-parameter)

Figure 90 shows the influence of diameter *d*-parameter on the ultimate resistance of the members. With increasing diameter, the ultimate strength of the members is increased significantly. This happens as expected since the area of cross section increases. An average increase of 88.1% and 57.2% on the ultimate load is experienced by member if diameter changing from 500 to 700*mm* and 700 to 900*mm*, respectively. Likewise, the displacement experiences an increase with increasing diameter which shows higher ductility.

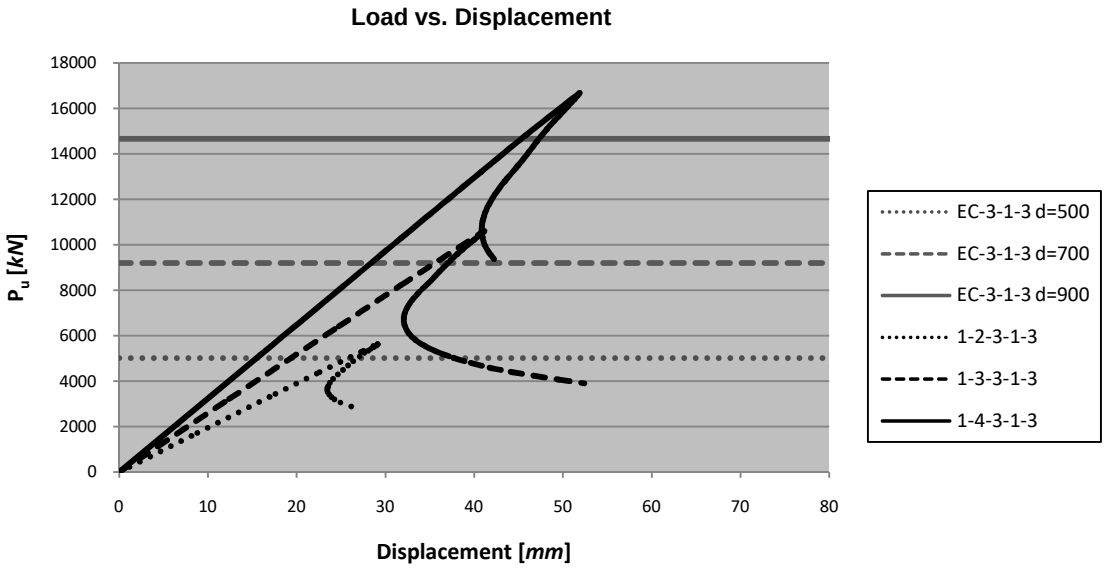


Figure 91. Load-displacement curve of numerical analysis for different diameters

Effect of *d*-parameter on the normalized resistance (P_u/P_{yg}) is shown in Figure 91. Six models were taken in comparison. The curve shows that diameter change has little influence on the normalized resistance. It means that by changing the diameter, reduction factor on ultimate cross-section resistance due to distortional buckling deviate negligibly. An average increase of 0.64% on normalized resistance occurs when diameter changing from 500 to 700*mm*, while average decrease of 2.1% occurs when diameter changing from 700 to 900*mm*.

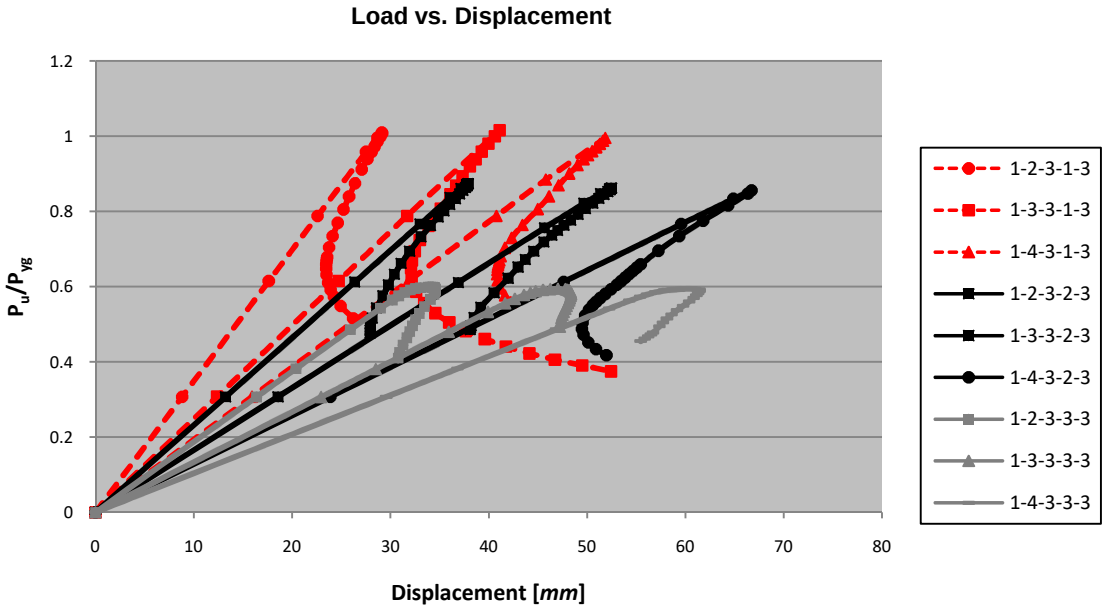


Figure 92. Load-displacement curve with normalized resistance (diff. diameters)

In order to make a uniform comparison of the normalized resistance between the parameters, a graph of normalized strength (P_u/P_{yg}) versus length represented by global slenderness($\bar{\lambda}$) was made. Figure 92 shows that different diameters have no significance on normalized resistance. The effect is less in the members in models with larger length or higher global slenderness.

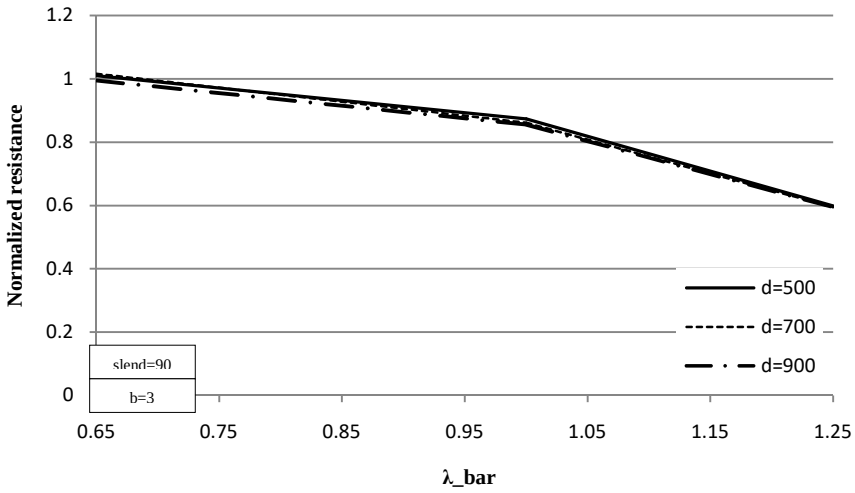


Figure 93.Normalized resistance vs. length for models with different diameters

4.3.2.2.2 Thickness (t-parameter)

Thickness parameter was calculated from cross-section slenderness input variable (*slend*) in the automation Python script. Therefore, this parameter depend on diameter (*d/t* ratio), and since two values of cross-section slenderness were taken, i.e. *slend* = 90 and 110, then there are two different thicknesses for each diameter or six different thickness in total. Models diameter 700*mm* with thickness 12 and 10*mm* were taken for comparison.

Figure 93 shows the influence of thickness *t*-parameter on the ultimate resistance of the members. With decreasing thickness, the ultimate strength of the members is also decreased. Alike *d*-parameter, this happens since thickness directly related to the area property of cross section. An average decrease of 21.5% on the ultimate load is experienced by member if thickness changing from 12 to 10*mm*. Likewise, the displacement experiences a decrease with decreasing thickness.

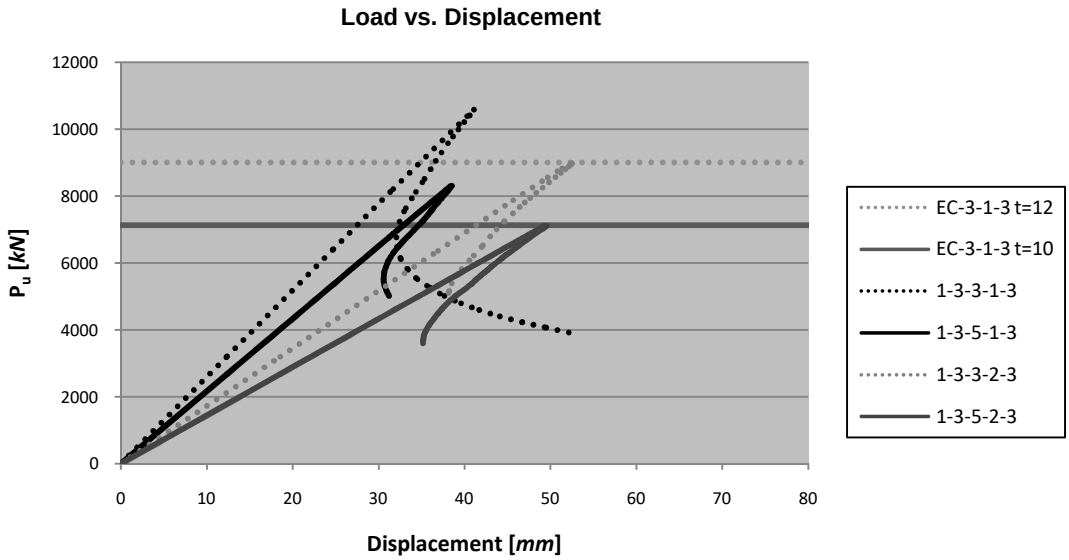


Figure 94. Load-displacement curve of numerical analysis for different thicknesses

Effect of *t*-parameter on the normalized resistance (P_u/P_{yg}) is shown in Figure 94. Six models were taken in comparison. The curve shows similar trend as the d-parameter, changing thicknesses have

little influence on the normalized resistance. By changing the thickness, reduction factor on ultimate cross-section resistance due to distortional buckling deviate negligibly. An average decrease of 6.3% on normalized resistance occurs when thickness changing from 12 to 10*mm*, while average decrease of 5.5% occurs with thickness from 12 to 10*mm* for models with larger length ($\bar{\lambda}$=1).

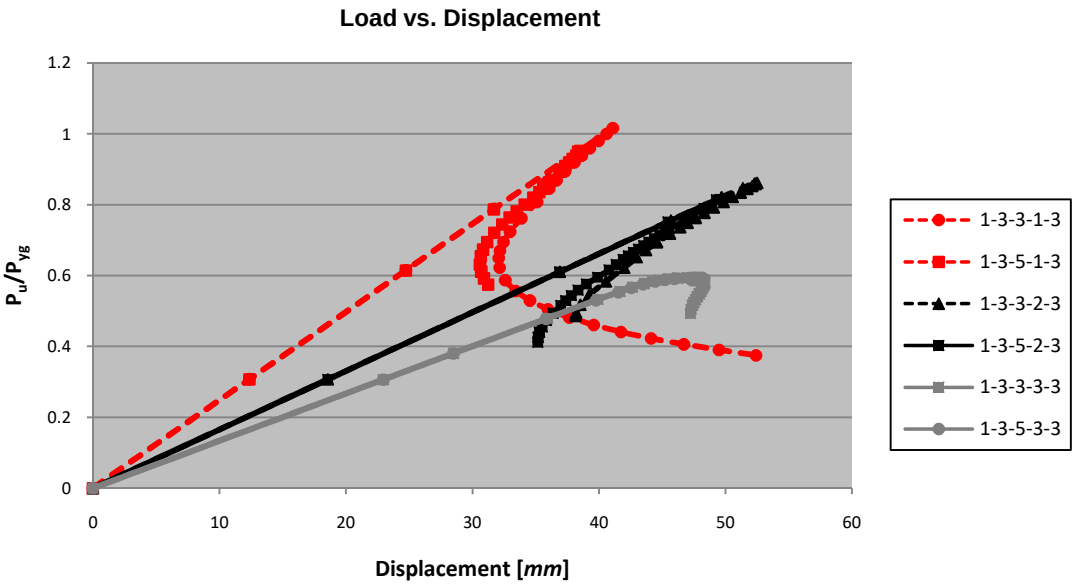


Figure 95.Load-displacement curve with normalized resistance (diff. thickness)

Figure 95 shows normalized strength (P_u/P_{yg}) versus length represented by global slenderness ($\bar{\lambda}$) for models with different thicknesses. As it can be seen the thickness change has little significance on normalized resistance. The effect is less in the members in models with larger length or higher global slenderness.

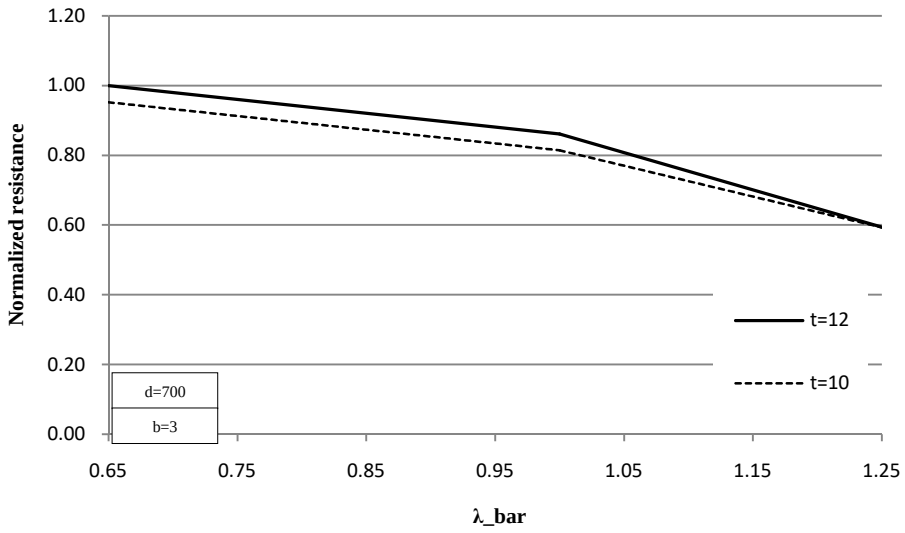


Figure 96. Normalized resistance vs. length for models with different thicknesses

4.3.2.2.3 Length ($\bar{\lambda}$-parameter)

Length parameter was calculated from input variable lambda, which is the global slenderness of the member. Three values were taken for slenderness, i.e. 0.65, 1.0, and 1.25, which in this comparison corresponds to length of 18106*mm*, 27048*mm*, and 33435*mm*, respectively. Models with diameter 500mm, bolt spacing ratio of *4*, and cross-section slenderness of 110 were used. As it can be seen in graph of Figure 96, more slender member will give less ultimate resistance. It is also noted that model with high global slenderness, i.e. 1-2-5-3-4 has ultimate strength much lower than the resistance calculated by EC1993-1-3 due to interaction of distortional mode with flexural mode. An average decrease of 9.4% and 22.9% on the ultimate load is experienced by member when $\bar{\lambda}$ changing from 0.65 to 1.0 and from 1.0 to 1.25, respectively. In contrary, the displacement experiences an increase with increasing slenderness.

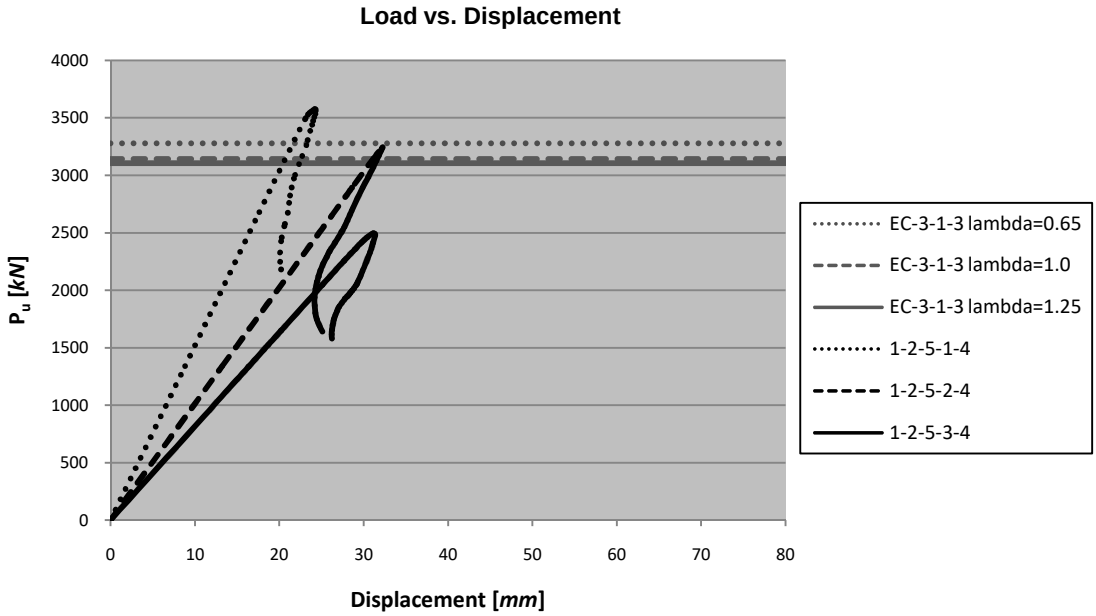


Figure 97. Load-displacement curve of numerical analysis for different lambda

Figure 97 shows effect of $\bar{\lambda}$-parameter on the normalized resistance (P_u/P_{yg}) due to change of global slenderness. Six models were taken in comparison. The curve shows similar trend as the other parameters, however the difference is visible. Changing length or slenderness has significant influence on the normalized resistance. By changing the slenderness, reduction factor on ultimate cross-section resistance due to distortional buckling deviate considerably. An average decrease of 9.4% on normalized resistance occurs when slenderness changing from 0.65 to 1.0, while average decrease of 22.9% occurs with changing slenderness from 1.0 into 1.25. A significant drop exists when the slenderness is above 1.0.

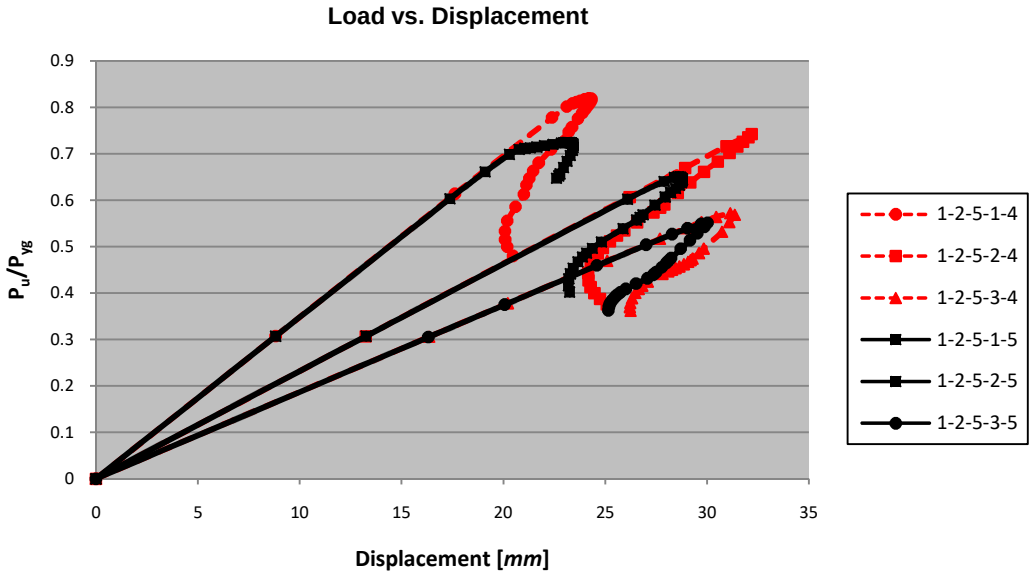


Figure 98. Load-displacement curve with normalized resistance (diff. global slenderness)

Figure 98 shows normalized strength (P_u/P_{yg}) versus global slenderness ($\bar{\lambda}$) for models with different lengths. As it can be seen the slenderness change has quite significant influence on normalized resistance, especially when the slenderness is above 1.0. The effect is more when the global slenderness is higher.

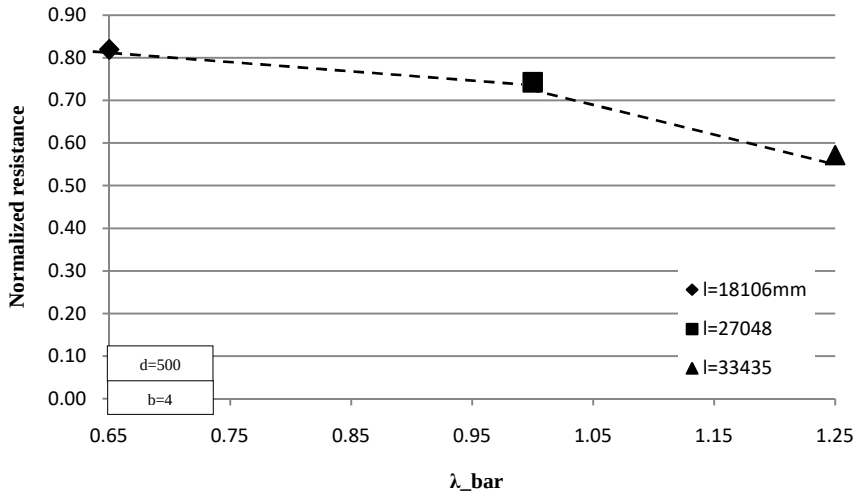


Figure 99. . Normalized resistance vs. length for models with different lengths

4.3.2.2.4 Bolt spacing (b-parameter)

Bolt spacing parameter as the *s/d* ratio was taken with three different values, i.e. 3, 4, and 5. This parameter is expected to give significant influence on the global behaviour and strength of the member since the stiffness of lip connection provided by these bolts determines the buckling mode of the section. Models with diameter 900mm, cross-section slenderness of 90, and global slenderness of 0.65 were used for comparison. Figure 99 shows that with denser bolts or closer bolt spacing, higher

ultimate resistance will be achieved. An average increase of 12.6% and 13.3% on the ultimate load is experienced by member for changing bolt ratio from 3 to 4 and from 4 to 5, respectively. Similarly, the displacement experiences an increase with closer bolt spacing.

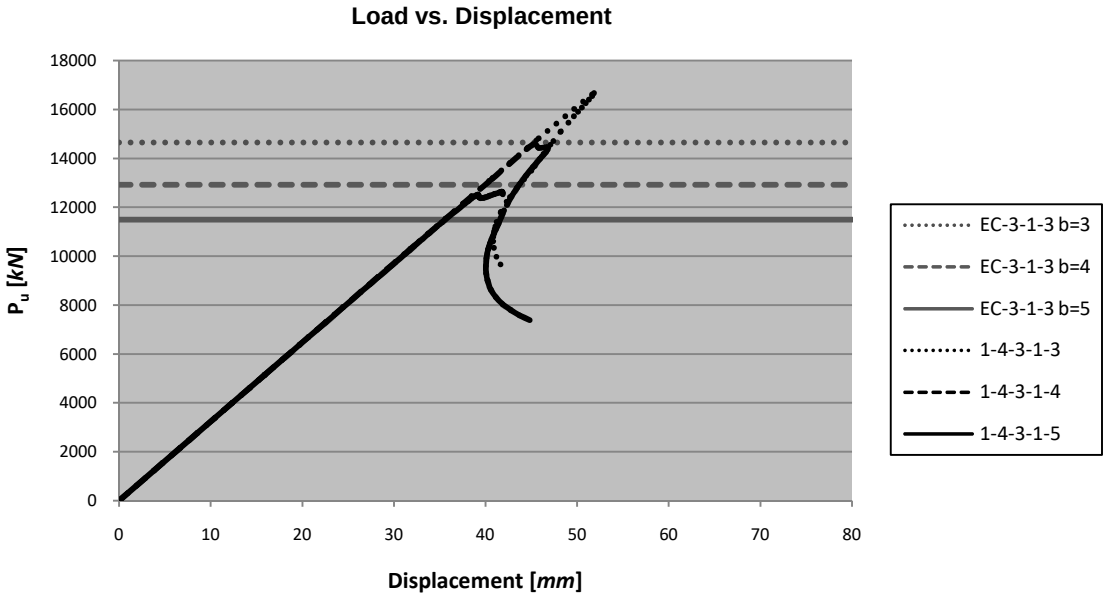


Figure 100. Load-displacement curve of numerical analysis for different bolt spacing

Effect of *b*-parameter on the normalized resistance (P_u/P_{yg}) is shown in Figure 100. Nine models were taken in comparison. Similarly to $\bar{\lambda}$-parameters, the curve shows significant change of normalized resistance by shifting the bolt spacing. By changing the bolt spacing, reduction factor on ultimate cross-section resistance due to distortional buckling deviates considerably. An average decrease of 12.6% on normalized resistance occurs when bolt spacing ratio changing from 3 to 4, while average decrease of 13.3% occurs with changing bolt spacing ratio from 4 to 5. The changes are in proportion with the ultimate load changes.

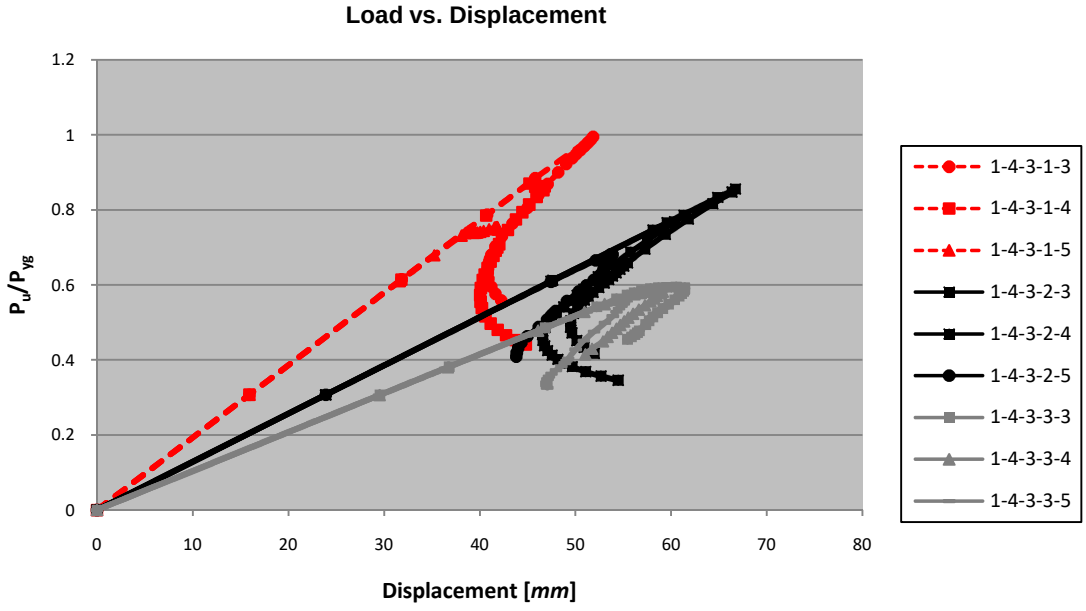


Figure 101. Load-displacement curve with normalized resistance (diff. bolt spacing)

Figure 101 shows normalized strength (P_u/P_{yg}) versus global slenderness ($\bar{\lambda}$) for models with different bolt spacing ratio. As it can be seen in the graph, the change of bolt spacing gives quite significant influence on normalized resistance. The effect is less in the members in models with larger length or higher global slenderness.

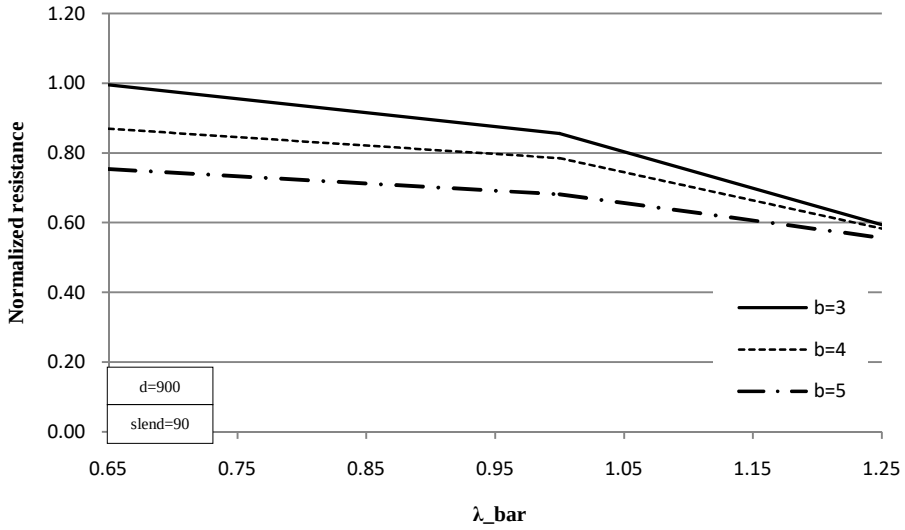


Figure 102. Normalized resistance vs. length for models with different bolt spacings

4.3.3 Analytical analysis for the proposed columns according to the standard rules EN1993-1-3 and EN1993-1-6

In this section, the results from numerical analysis of the models used in parametric studies were compared and verified to the analytical calculations according to EN1993-1-3 *General rules for cold-formed members and sheeting* and EN1993-1-6 *Strength and stability of shell structures.* The numerical results were compared to these two design codes with consideration that the expected behaviour of the proposed columns is in between of plates and shell structural element. Due to large amount of models, calculations were performed with Python script automation; detail of the script can be found in the Annex A.5.

4.3.3.1 Analytical analysis according to EN1993-1-3

EN1993-1-3 considers the studied semi-closed polygonal sections as built-up member composed of plates. Each adjacent edge of the polygonal was considered as simply supported connected plate.

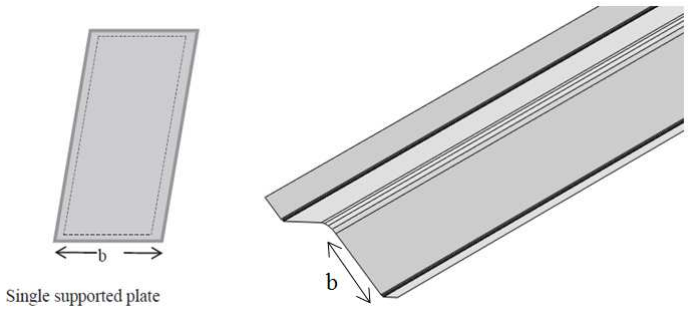


Figure 103. Simply supported plates compose the polygonal cross-section

The procedure of calculation is described in chapter 2.3.7 and the complete results are presented in the Annex. The determination of ultimate resistance of the member includes cross-section resistance and buckling resistance. Calculation of cross-section resistance covers the local and distortional buckling mode through reduction factor, namely ρ and χ_d. Reduction factor due to local buckling is applied to the plane plate according to the class of cross-section, refer to EC1993-1-5 [**10**]. Meanwhile, reduction factor due to distortional buckling is applied to the stiffener element either edge or intermediate stiffener, and taken into account for flexural buckling of the stiffener. The reduction factor due to local and or distortional buckling is used as the reduction of the cross-section area. Calculation of buckling resistance of cold-formed member includes flexural (F), torsional (T), and flexural torsional (FT) buckling. The lowest reduction factor between these three buckling modes will govern and used as reduction in buckling resistance calculation.

Part 3.2 and 5.1 of the code rule out the requirement for geometrical and material properties verification including material increased average yield strength (f_{ya}) and influence of rounded corners, respectively. These two requirements were taken into account and calculated in this analysis accordingly. Average increased average yield strength was not applicable since the effective areas of cross section were used in the calculation, while influence of rounded corners can be neglected due to the using of plane elements along corners during the calculation. Detail of results for this calculation can be seen in the Annex D.

Cross section classification was carried out according to EN1993-1-1 by considering the polygonal profiles are composed of simply supported plates. All cross-section models fall in class 3 cross-section. Double check was done in MATLAB by calculating class of each plate and took the lowest class as the cross-section class, and by calculating the effective area of each plate then sum up them as the cross-section effective area. Both methods have agreement in the results. Complete script for calculating the classification of cross section and effective area can be seen in the MATLAB script for profiles *polygoner.m* line 81 – 133 in Annex A.2.

In the analytical calculation, lips along the profiles are considered as stiffeners and refer to EC1993-1-3 part 5.5.3 plane elements with edge or intermediate stiffeners. It takes into account the assumption that stiffener behaves as a compression element with continuous partial restraint, with a spring stiffness and the flexural stiffness of the adjacent plane elements. This stiffener subjects to distortional buckling mode. Reduction factor due to distortional buckling was calculated based on stiffness and local slenderness of the stiffener. The procedure included iterative process to obtain refined and less conservative result by calculating the effective width with a reduced compressive stress $\sigma_{com,Ed} = \chi_d f_{yb}/\gamma_{M0}$with χ_d taken from previous iteration. In this study, iteration was done until the third step where enough convergence was achieved, $\chi_{d,n} \approx \chi_{d,(n-1)}$ but $\chi_{d,n} \leq \chi_{d,(n-1)}$.

Since all cross section are in class 3 then there is no local buckling mode of failure, which corresponds to the numerical results. Therefore, reduction of cross-section area only comes from distortional buckling, which applies to the thickness of stiffener area plus the effective portions of the adjacent plane element.

Table 12 shows the result of analytical calculation according to EC1993-1-3 for cross-section resistance and comparison between numerical and analytical results.

Table 12. Analytical calculation result of the cross-section resistance for studied columns according to EN1993-1-3 and comparison of the resistance to the FE results

(a) *b*-parameter=3

Model ID	d	t	λ	s/d	Class	Area	Cross section resistance							FE analysis	FE/ EC3-1-3
							Local	Eff. area local	Dist. slend	Dist.	Eff. area dist.	$N_{c,Rd}$			
						A_g	ρ_b	$A_{eff,loc}$	λ_d	χ_{dist}	$A_{eff,dis}$				
	[mm]	[mm]	[-]	[-]	[-]	[cm^2]	[-]	[cm^2]	[-]	[-]	[cm^2]	[kN]	[kN]	[-]
1 2 3 1 3	500	9	0.65		3	157.0	1.0	157.0	0.79	0.898	141.0	5003.9	5625.7	1.12
1 3 3 1 3	700	12			3	293.5	1.0	293.5	0.81	0.882	258.9	9192.4	10583.1	1.15
1 4 3 1 3	900	15			3	472.0	1.0	472.0	0.82	0.874	412.8	14653.3	16670.0	1.14
1 2 5 1 3	500	7			3	123.0	1.0	123.0	0.85	0.853	104.9	3724.3	4132.5	1.11
1 3 5 1 3	700	10			3	245.8	1.0	245.8	0.86	0.851	209.2	7425.1	8309.6	1.12
1 4 5 1 3	900	13			3	410.7	1.0	410.7	0.86	0.850	349.3	12398.7	13828.6	1.12
1 2 3 2 3	500	9	1	3	3	157.0	1.0	157.0	0.81	0.884	138.8	4927.1	4865.7	0.99
1 3 3 2 3	700	12			3	293.5	1.0	293.5	0.83	0.868	254.6	9038.6	8974.8	0.99
1 4 3 2 3	900	15			3	472.0	1.0	472.0	0.85	0.859	405.4	14391.9	14330.1	1.00
1 2 5 2 3	500	7			3	123.0	1.0	123.0	0.88	0.836	102.8	3650.2	3665.2	1.00
1 3 5 2 3	700	10			3	245.8	1.0	245.8	0.88	0.827	203.2	7213.2	7105.3	0.99
1 4 5 2 3	900	13			3	410.7	1.0	410.7	0.88	0.833	342.2	12148.3	12124.6	1.00
1 2 3 3 3	500	9	1.25		3	157.0	1.0	157.0	1.28	0.542	85.1	3020.0	3335.7	1.10
1 3 3 3 3	700	12			3	293.5	1.0	293.5	1.29	0.540	158.6	5628.8	6192.9	1.10
1 4 3 3 3	900	15			3	472.0	1.0	472.0	1.29	0.539	254.2	9024.0	9958.4	1.10
1 2 5 3 3	500	7			3	123.0	1.0	123.0	1.28	0.543	66.7	2369.0	2606.7	1.10
1 3 5 3 3	700	10			3	245.8	1.0	245.8	1.29	0.540	132.7	4710.1	5177.7	1.10
1 4 5 3 3	900	13			3	410.7	1.0	410.7	1.29	0.538	221.0	7846.1	8656.9	1.10

(b) *b*-parameter=4

Model ID	d	t	λ	s/d	Class	Area	Cross section resistance							FE analysis	FE/ EC3-1-3
							Local	Eff. area local	Dist. slend	Dist.	Eff. area dist.	$N_{c,Rd}$			
						A_g	ρ_b	$A_{eff,loc}$	λ_d	χ_{dist}	$A_{eff,dis}$				
	[mm]	[mm]	[-]	[-]	[-]	[cm^2]	[-]	[cm^2]	[-]	[-]	[cm^2]	[kN]	[kN]	[-]
1 2 3 1 4	500	9	0.65		3	157.0	1.0	157.0	0.92	0.808	126.8	4502.4	4982.5	1.11
1 3 3 1 4	700	12			3	293.5	1.0	293.5	0.95	0.785	230.4	8178.2	9053.7	1.11
1 4 3 1 4	900	15			3	472.0	1.0	472.0	0.97	0.771	364.1	12924.9	14571.6	1.13
1 2 5 1 4	500	7			3	123.0	1.0	123.0	0.99	0.751	92.3	3278.4	3576.7	1.09
1 3 5 1 4	700	10			3	245.8	1.0	245.8	1.00	0.745	183.2	6502.9	7358.9	1.13
1 4 5 1 4	900	13			3	410.7	1.0	410.7	1.01	0.741	304.5	10809.4	11788.0	1.09
1 2 3 2 4	500	9	1	4	3	157.0	1.0	157.0	0.95	0.784	123.0	4368.0	4555.0	1.04
1 3 3 2 4	700	12			3	293.5	1.0	293.5	0.99	0.758	222.3	7893.1	8128.1	1.03
1 4 3 2 4	900	15			3	472.0	1.0	472.0	1.01	0.742	350.1	12426.9	13158.7	1.06
1 2 5 2 4	500	7			3	123.0	1.0	123.0	1.04	0.720	88.5	3143.1	3241.9	1.03
1 3 5 2 4	700	10			3	245.8	1.0	245.8	1.05	0.713	175.2	6221.0	6373.9	1.02
1 4 5 2 4	900	13			3	410.7	1.0	410.7	1.05	0.708	290.8	10322.0	10701.6	1.04
1 2 3 3 4	500	9	1.25		3	157.0	1.0	157.0	0.96	0.777	122.0	4332.5	3269.5	0.75
1 3 3 3 4	700	12			3	293.5	1.0	293.5	1.00	0.750	220.2	7817.0	6020.5	0.77
1 4 3 3 4	900	15			3	472.0	1.0	472.0	1.02	0.734	346.3	12293.2	9777.5	0.80
1 2 5 3 4	500	7			3	123.0	1.0	123.0	1.05	0.712	87.5	3107.0	2496.8	0.80
1 3 5 3 4	700	10			3	245.8	1.0	245.8	1.06	0.704	173.1	6146.6	4985.4	0.81
1 4 5 3 4	900	13			3	410.7	1.0	410.7	1.07	0.699	287.0	10189.7	8400.3	0.82

(c) *b*-parameter=5

Model ID	d	t	λ	s/d	Class	Area	Cross section resistance							FE analysis	FE/ EC3-1-3
							Local	Eff. area local	Dist. slend	Dist.	Eff. area dist.	$N_{c,Rd}$			
						A_g	ρ_b	$A_{eff,loc}$	λ_d	χ_{dist}	$A_{eff,dis}$				
	[mm]	[mm]	[-]	[-]	[-]	[cm^2]	[-]	[cm^2]	[-]	[-]	[cm^2]	[kN]	[kN]	[-]
1 2 3 1 5	500	9	0.65		3	157.0	1.0	157.0	1.02	0.733	115.0	4082.8	4531.8	1.11
1 3 3 1 5	700	12			3	293.5	1.0	293.5	1.06	0.704	206.6	7332.6	8140.8	1.11
1 4 3 1 5	900	15			3	472.0	1.0	472.0	1.08	0.686	323.7	11491.3	12630.0	1.10
1 2 5 1 5	500	7			3	123.0	1.0	123.0	1.11	0.667	82.0	2912.5	3161.9	1.09
1 3 5 1 5	700	10			3	245.8	1.0	245.8	1.12	0.657	161.6	5736.4	6417.5	1.12
1 4 5 1 5	900	13			3	410.7	1.0	410.7	1.05	0.714	293.4	10415.4	10664.4	1.02
1 2 3 2 5	500	9	1	5	3	157.0	1.0	157.0	1.07	0.698	109.6	3890.3	4126.2	1.06
1 3 3 2 5	700	12			3	293.5	1.0	293.5	1.11	0.664	194.9	6917.5	7400.3	1.07
1 4 3 2 5	900	15			3	472.0	1.0	472.0	1.15	0.642	303.0	10758.1	11412.8	1.06
1 2 5 2 5	500	7			3	123.0	1.0	123.0	1.17	0.623	76.6	2720.3	2840.9	1.04
1 3 5 2 5	700	10			3	245.8	1.0	245.8	1.19	0.610	150.0	5323.7	5661.1	1.06
1 4 5 2 5	900	13			3	410.7	1.0	410.7	1.20	0.600	246.6	8752.7	9216.9	1.05
1 2 3 3 5	500	9	1.25		3	157.0	1.0	157.0	1.08	0.688	108.1	3836.6	3176.2	0.83
1 3 3 3 5	700	12			3	293.5	1.0	293.5	1.13	0.653	191.5	6798.7	5901.9	0.87
1 4 3 3 5	900	15			3	472.0	1.0	472.0	1.16	0.629	297.1	10545.4	9320.6	0.88
1 2 5 3 5	500	7			3	123.0	1.0	123.0	1.19	0.610	75.1	2664.5	2405.3	0.90
1 3 5 3 5	700	10			3	245.8	1.0	245.8	1.21	0.597	146.7	5206.3	4683.0	0.90
1 4 5 3 5	900	13			3	410.7	1.0	410.7	1.22	0.586	240.5	8538.3	7913.5	0.93

The ultimate resistance of cross-section from the finite element analysis showed a good agreement with EN-1993 part 1-3. However, some numerical models have a disagreement and show unsafe predictions compared to the design standard. Cluster of models with global slenderness $\bar{\lambda}$=1,25 and bolt spacing *b*=3 and 4 exhibit considerably lower resistance than the design standard. Examination showed that these models are all models failed in interaction mode, i.e. distortional-flexural interaction, as shown in Figure 84. Therefore, it can be concluded that the analytical calculation based on EC1993-1-3 performed in this section corresponds to the numerical analysis described in chapter 4.3.2; whereas the deviations due to the interaction mode cannot be captured by EN 1993-1-3.

Besides cross-sections resistance, design standard EC1993-1-3 requires calculation of buckling resistance. In this case, flexural buckling was expected to govern between the three buckling modes, i.e. flexural, torsional, and flex-torsional buckling since the semi-closed connection provide high torsional stiffness on the cross-section, as shown in table of cross-section properties and chapter 3.2.1 verification of elastic buckling analysis. Calculation of flexural buckling resistance was done according to EC1993-1-1. Effective area taken from cross-section resistance calculation was used when calculating the global slenderness. Buckling class b=0.34 was used, as recommended by EC1993-1-3. The reduction factor is mainly determined by those two above-mentioned properties. As for torsional buckling, the expression in EC1993-1-3 was used. Principally, the reduction factor for torsional and flexural torsional buckling is calculated by substituting elastic critical stress in in the equations of flexural buckling with torsional and flexural-torsional critical stress. The lowest reduction factor will be the governing buckling mode and then used in the buckling resistance formulae.

$$N_{b,Rd} = \chi . A_{eff} . \frac{f_{yb}}{\gamma_{M1}}$$

Table 13. Analytical calculation result of the buckling resistance for studied columns according to EN1993-1-3 and comparison of the resistance to the FE results

(a) *b*-parameter=3

Model ID	d	t	λ	s/d	Class	Area	Buckling resistance					FE analysis	FE/ EC3-1-3
							Eff. Area	Flex.	Tor.	Flex-Tor	$N_{b,Rd}$		
						A_g	A_{eff}	χ_F	χ_T	χ_{FT}			
	[mm]	[mm]	[-]	[-]	[-]	[cm^2]	[cm^2]	[-]	[-]	[-]	[kN]	[kN]	[-]
1 2 3 1 3	500	9	0.65	3	3	157.0	141.0	0.819	1.0	0.819	4097.5	5625.7	1.37
1 3 3 1 3	700	12			3	293.5	258.9	0.820	1.0	0.820	7536.4	10583.1	1.40
1 4 3 1 3	900	15			3	472.0	412.8	0.820	1.0	0.820	12022.1	16670.0	1.39
1 2 5 1 3	500	7			3	123.0	104.9	0.824	1.0	0.824	3069.7	4132.5	1.35
1 3 5 1 3	700	10			3	245.8	209.2	0.824	1.0	0.824	6116.7	8309.6	1.36
1 4 5 1 3	900	13			3	410.7	349.3	0.824	1.0	0.824	10210.9	13828.6	1.35
1 2 3 2 3	500	9	1	3	3	157.0	138.8	0.611	1.0	0.611	3011.8	4865.7	1.62
1 3 3 2 3	700	12			3	293.5	254.6	0.613	1.0	0.613	5541.8	8974.8	1.62
1 4 3 2 3	900	15			3	472.0	405.4	0.614	1.0	0.614	8839.9	14330.1	1.62
1 2 5 2 3	500	7			3	123.0	102.8	0.621	1.0	0.621	2268.5	3665.2	1.62
1 3 5 2 3	700	10			3	245.8	203.2	0.621	1.0	0.621	4476.4	7105.3	1.59
1 4 5 2 3	900	13			3	410.7	342.2	0.620	1.0	0.620	7533.4	12124.6	1.61
1 2 3 3 3	500	9	1.25		3	157.0	85.1	0.466	1.0	0.466	1408.2	3335.7	2.37
1 3 3 3 3	700	12			3	293.5	158.6	0.468	1.0	0.468	2635.6	6192.9	2.35
1 4 3 3 3	900	15			3	472.0	254.2	0.469	1.0	0.469	4235.6	9958.4	2.35
1 2 5 3 3	500	7			3	123.0	66.7	0.477	1.0	0.477	1129.9	2606.7	2.31
1 3 5 3 3	700	10			3	245.8	132.7	0.476	1.0	0.476	2242.1	5177.7	2.31
1 4 5 3 3	900	13			3	410.7	221.0	0.476	1.0	0.476	3731.1	8656.9	2.32

(b) *b*-parameter=4

Model ID	d	t	λ	s/d	Class	Area	Buckling resistance					FE analysis	FE/ EC3-1-3
							Eff. Area	Flex.	Tor.	Flex-Tor	$N_{b,Rd}$		
						A_g	A_{eff}	χ_F	χ_T	χ_{FT}			
	[mm]	[mm]	[-]	[-]	[-]	[cm^2]	[cm^2]	[-]	[-]	[-]	[kN]	[kN]	[-]
1 2 3 1 4	500	9	0.65	4	3	157.0	126.8	0.819	1.0	0.819	3686.9	4982.5	1.35
1 3 3 1 4	700	12			3	293.5	230.4	0.820	1.0	0.820	6704.9	9053.7	1.35
1 4 3 1 4	900	15			3	472.0	364.1	0.820	1.0	0.820	10604.0	14571.6	1.37
1 2 5 1 4	500	7			3	123.0	92.3	0.824	1.0	0.824	2702.2	3576.7	1.32
1 3 5 1 4	700	10			3	245.8	183.2	0.824	1.0	0.824	5357.0	7358.9	1.37
1 4 5 1 4	900	13			3	410.7	304.5	0.824	1.0	0.824	8902.0	11788.0	1.32
1 2 3 2 4	500	9	1	4	3	157.0	123.0	0.611	1.0	0.611	2670.0	4555.0	1.71
1 3 3 2 4	700	12			3	293.5	222.3	0.613	1.0	0.613	4839.5	8128.1	1.68
1 4 3 2 4	900	15			3	472.0	350.1	0.614	1.0	0.614	7632.9	13158.7	1.72
1 2 5 2 4	500	7			3	123.0	88.5	0.621	1.0	0.621	1953.3	3241.9	1.66
1 3 5 2 4	700	10			3	245.8	175.2	0.621	1.0	0.621	3860.7	6373.9	1.65
1 4 5 2 4	900	13			3	410.7	290.8	0.620	1.0	0.620	6400.9	10701.6	1.67
1 2 3 3 4	500	9	1.25		3	157.0	122.0	0.466	1.0	0.466	2020.3	3269.5	1.62
1 3 3 3 4	700	12			3	293.5	220.2	0.468	1.0	0.468	3660.2	6020.5	1.64
1 4 3 3 4	900	15			3	472.0	346.3	0.469	1.0	0.469	5770.1	9777.5	1.69
1 2 5 3 4	500	7			3	123.0	87.5	0.477	1.0	0.477	1481.9	2496.8	1.68
1 3 5 3 4	700	10			3	245.8	173.1	0.476	1.0	0.476	2926.0	4985.4	1.70
1 4 5 3 4	900	13			3	410.7	287.0	0.476	1.0	0.476	4845.6	8400.3	1.73

(c) *b*-parameter=5

Model ID	d	t	λ	s/d	Class	Area	Buckling resistance					FE analysis	FE/ EC3-1-3
							Eff. Area	Flex.	Tor.	Flex-Tor	$N_{b,Rd}$		
						A_g	A_{eff}	χ_F	χ_T	χ_{FT}			
	[mm]	[mm]	[-]	[-]	[-]	[cm^2]	[cm^2]	[-]	[-]	[-]	[kN]	[kN]	[-]
1 2 3 1 5	500	9	0.65		3	157.0	115.0	0.819	1.0	0.819	3343.3	4531.8	1.36
1 3 3 1 5	700	12			3	293.5	206.6	0.820	1.0	0.820	6011.6	8140.8	1.35
1 4 3 1 5	900	15			3	472.0	323.7	0.820	1.0	0.820	9427.9	12630.0	1.34
1 2 5 1 5	500	7			3	123.0	82.0	0.824	1.0	0.824	2400.6	3161.9	1.32
1 3 5 1 5	700	10			3	245.8	161.6	0.824	1.0	0.824	4725.6	6417.5	1.36
1 4 5 1 5	900	13			3	410.7	293.4	0.824	1.0	0.824	8577.5	10664.4	1.24
1 2 3 2 5	500	9	1	5	3	157.0	109.6	0.611	1.0	0.611	2378.1	4126.2	1.74
1 3 3 2 5	700	12			3	293.5	194.9	0.613	1.0	0.613	4241.3	7400.3	1.74
1 4 3 2 5	900	15			3	472.0	303.0	0.614	1.0	0.614	6607.9	11412.8	1.73
1 2 5 2 5	500	7			3	123.0	76.6	0.621	1.0	0.621	1690.5	2840.9	1.68
1 3 5 2 5	700	10			3	245.8	150.0	0.621	1.0	0.621	3303.8	5661.1	1.71
1 4 5 2 5	900	13			3	410.7	246.6	0.620	1.0	0.620	5427.7	9216.9	1.70
1 2 3 3 5	500	9	1.25		3	157.0	108.1	0.466	1.0	0.466	1789.0	3176.2	1.78
1 3 3 3 5	700	12			3	293.5	191.5	0.468	1.0	0.468	3183.3	5901.9	1.85
1 4 3 3 5	900	15			3	472.0	297.1	0.469	1.0	0.469	4949.7	9320.6	1.88
1 2 5 3 5	500	7			3	123.0	75.1	0.477	1.0	0.477	1270.9	2405.3	1.89
1 3 5 3 5	700	10			3	245.8	146.7	0.476	1.0	0.476	2478.4	4683.0	1.89
1 4 5 3 5	900	13			3	410.7	240.5	0.476	1.0	0.476	4060.3	7913.5	1.95

Table 13 shows the results of buckling resistance calculation according to EC1993-1-3 and comparison with numerical results. As expected, the governing global buckling mode was flexural type. No reduction on torsional buckling mode proves that the torsional stiffness of the cross section was fully (or almost fully) developed, and moreover shows the effectiveness of lip connection that form semi-closed cross-section. As for comparison, it can be seen that the ultimate resistance from FE analysis were much higher than the EC-1993-1-3 buckling resistance. The design standard underpredicted the strengths of all models since the failure modes that occurred were not global flexural type. From these analyses it is known that this type of cross-section, with the specified local and global slenderness, seemed much more prominent for distortional failure mode.

It is worth noting that for this type of cross-section, adopting the design standard without knowing the exact failure mode can result in too conservative predictions. Local slenderness comes from the bolt spacing which significantly determine the failure mode and hence the ultimate strength of the member, is not included in the current design standard.

4.3.3.2 Analytical analysis according to EN1993-1-6

EN1993-1-6 was used to see the closeness of this type of cross-section to the behaviour of shell structures and also the applicability of this design standard for this type of sections. By using EN1993-1-6, the studied columns were considered as tubular cross section which means fully closed and fully rigid connected. The studied semi-closed polygonal sections were then classified with refer to Table 5.2 sheet 3 of EC1993-1-1. Based on this table, the limit for class 3 cross-sections is $d/t \leq 90\varepsilon^2$, with d/ ε^2t known as cross-section slenderness. Since two values of cross-section slenderness, i.e. 90 and 110 were used in this study then some cross-sections will be class 4.

The procedure of calculation based on EC1993-1-6 is described in chapter 2.3.8 and the workflow can be seen in Figure 30.Ultimate resistance calculation was intended to investigate loss of stability under compressive membrane or shear membrane stresses in the shell wall, as expected will govern in this studied columns. The buckling limit state (LS3) was calculated for this purpose.

Design resistances for stress components are obtained using buckling reduction factors χ taking into account:

- Imperfections depending on the Fabrication Tolerance Quality Class
- Boundary Conditions of the cylindrical shell

The partial safety factor may be defined in the National Annex. The recommended value is $\gamma_{M1} = 1.1$. The buckling reduction factors are determined as a function of the relative slenderness of the shell, defined for different stress components, i.e. meridional, circumferential and shear. The elastic critical buckling stresses were obtained using appropriate expression in Annex D of EC1993-1-6. In this study, the parameter C_{xb} was taken equal to 1.0 as a conservative value for long equivalent cylinders, while the characteristic imperfection amplitude Δw_k was calculated based on fabrication quality parameter *Q*, which is taken as Class C (normal), equal to 16. Other parameters should be taken as $\bar{\lambda}_{x0}$=0.2, β=0.6, and η=1.0. Detail calculation of analytical based on EC1993-1-6 can be seen in Python script in Annex A.5. Results of the analytical calculation are presented in Table 18. For cross-sections with class 3, the design resistance was calculated as the full effective cross-section resistance, as per EC1993-1-1.

Table 14. Analytical calculation result of the design resistance for studied columns according to EN1993-1-6 and comparison of the resistance to the FE results

(a) *b*-parameter=3

Model ID	d	t	l	d/t	λ	s/d	Real cs-slend	Class	Area	Design resistance	FE analysis	FE/ EC3-1-6
									A_g	NRd_shell		
	[mm]	[mm]	[mm]	[-]	[-]	[-]	[-]	[-]	[cm^2]	[kN]	[kN]	[-]
1 2 3 1 3	500	9	18098	55.6	0.65	1	83.9	3	157.0	5573.43	5625.7	1.01
1 3 3 1 3	700	12	25339	58.3			88.1	3	293.5	10418.6	10583.1	1.02
1 4 3 1 3	900	15	32580	60.0			90.6	4	472.0	13714.0	16670.0	1.22
1 2 5 1 3	500	7	18106	71.4			107.9	4	123.0	3495.4	4132.5	1.18
1 3 5 1 3	700	10	25347	70.0			105.7	4	245.8	7005.2	8309.6	1.19
1 4 5 1 3	900	13	32588	69.2			104.6	4	410.7	11721.7	13828.6	1.18
1 2 3 2 3	500	9	27035	55.6	1	3	83.9	3	157.0	5573.4	4865.7	0.87
1 3 3 2 3	700	12	37852	58.3			88.1	3	293.5	10418.6	8974.8	0.86
1 4 3 2 3	900	15	48669	60.0			90.6	4	472.0	13714.0	14330.1	1.04
1 2 5 2 3	500	7	27048	71.4			107.9	4	123.0	3495.4	3665.2	1.05
1 3 5 2 3	700	10	37865	70.0			105.7	4	245.8	7005.2	7105.3	1.01
1 4 5 2 3	900	13	48682	69.2			104.6	4	410.7	11721.7	12124.6	1.03
1 2 3 3 3	500	9	33419	55.6	1.25		83.9	3	157.0	5573.4	3335.7	0.60
1 3 3 3 3	700	12	46790	58.3			88.1	3	293.5	10418.6	6192.9	0.59
1 4 3 3 3	900	15	60161	60.0			90.6	4	472.0	13714.0	9958.4	0.73
1 2 5 3 3	500	7	33435	71.4			107.9	4	123.0	3495.4	2606.7	0.75
1 3 5 3 3	700	10	46806	70.0			105.7	4	245.8	7005.2	5177.7	0.74
1 4 5 3 3	900	13	60177	69.2			104.6	4	410.7	11721.7	8656.9	0.74

(b) *b*-parameter=4

Model ID	d	t	l	d/t	λ	s/d	Real cs-slend	Class	Area	Design resistance	FE analysis	FE/ EC3-1-6
									A_g	NRd_shell		
	[mm]	[mm]	[mm]	[-]	[-]	[-]	[-]	[-]	[cm^2]	[kN]	[kN]	[-]
1 2 3 1 4	500	9	18098	55.6	0.65	1 4	83.9	3	157.0	5573.4	4982.5	0.89
1 3 3 1 4	700	12	25339	58.3			88.1	3	293.5	10418.6	9053.7	0.87
1 4 3 1 4	900	15	32580	60.0			90.6	4	472.0	13714.0	14571.6	1.06
1 2 5 1 4	500	7	18106	71.4			107.9	4	123.0	3495.4	3576.7	1.02
1 3 5 1 4	700	10	25347	70.0			105.7	4	245.8	7005.2	7358.9	1.05
1 4 5 1 4	900	13	32588	69.2			104.6	4	410.7	11721.7	11788.0	1.01
1 2 3 2 4	500	9	27035	55.6	1		83.9	3	157.0	5573.4	4555.0	0.82
1 3 3 2 4	700	12	37852	58.3			88.1	3	293.5	10418.6	8128.1	0.78
1 4 3 2 4	900	15	48669	60.0			90.6	4	472.0	13714.0	13158.7	0.96
1 2 5 2 4	500	7	27048	71.4			107.9	4	123.0	3495.4	3241.9	0.93
1 3 5 2 4	700	10	37865	70.0			105.7	4	245.8	7005.2	6373.9	0.91
1 4 5 2 4	900	13	48682	69.2			104.6	4	410.7	11721.7	10701.6	0.91
1 2 3 3 4	500	9	33419	55.6	1.25		83.9	3	157.0	5573.4	3269.5	0.59
1 3 3 3 4	700	12	46790	58.3			88.1	3	293.5	10418.6	6020.5	0.58
1 4 3 3 4	900	15	60161	60.0			90.6	4	472.0	13714.0	9777.5	0.71
1 2 5 3 4	500	7	33435	71.4			107.9	4	123.0	3495.4	2496.8	0.71
1 3 5 3 4	700	10	46806	70.0			105.7	4	245.8	7005.2	4985.4	0.71
1 4 5 3 4	900	13	60177	69.2			104.6	4	410.7	11721.7	8400.3	0.72

(c) *b*-parameter=5

Model ID	d	t	l	d/t	λ	s/d	Real cs-slend	Class	Area	Design resistance	FE analysis	FE/ EC3-1-6
									A_g	NRd_shell		
	[mm]	[mm]	[mm]	[-]	[-]	[-]	[-]	[-]	[cm^2]	[kN]	[kN]	[-]
1 2 3 1 5	500	9	18098	55.6	0.65	1 5	83.9	3	157.0	5573.4	4531.8	0.81
1 3 3 1 5	700	12	25339	58.3			88.1	3	293.5	10418.6	8140.8	0.78
1 4 3 1 5	900	15	32580	60.0			90.6	4	472.0	13714.0	12630.0	0.92
1 2 5 1 5	500	7	18106	71.4			107.9	4	123.0	3495.4	3161.9	0.90
1 3 5 1 5	700	10	25347	70.0			105.7	4	245.8	7005.2	6417.5	0.92
1 4 5 1 5	900	13	32588	69.2			104.6	4	410.7	11721.7	10664.4	0.91
1 2 3 2 5	500	9	27035	55.6	1		83.9	3	157.0	5573.4	4126.2	0.74
1 3 3 2 5	700	12	37852	58.3			88.1	3	293.5	10418.6	7400.3	0.71
1 4 3 2 5	900	15	48669	60.0			90.6	4	472.0	13714.0	11412.8	0.83
1 2 5 2 5	500	7	27048	71.4			107.9	4	123.0	3495.4	2840.9	0.81
1 3 5 2 5	700	10	37865	70.0			105.7	4	245.8	7005.2	5661.1	0.81
1 4 5 2 5	900	13	48682	69.2			104.6	4	410.7	11721.7	9216.9	0.79
1 2 3 3 5	500	9	33419	55.6	1.25		83.9	3	157.0	5573.4	3176.2	0.57
1 3 3 3 5	700	12	46790	58.3			88.1	3	293.5	10418.6	5901.9	0.57
1 4 3 3 5	900	15	60161	60.0			90.6	4	472.0	13714.0	9320.6	0.68
1 2 5 3 5	500	7	33435	71.4			107.9	4	123.0	3495.4	2405.3	0.69
1 3 5 3 5	700	10	46806	70.0			105.7	4	245.8	7005.2	4683.0	0.67
1 4 5 3 5	900	13	60177	69.2			104.6	4	410.7	11721.7	7913.5	0.68

Table 14 shows the ultimate resistance based on the design standard and comparison with FE analysis. It shows the same trend as resistance calculated by EC1993-1-3, characterized by declining resistance with increasing member slenderness and bolt spacing. However, more models are in unsafe region when comparing the FE results with the analytical ones. This result gives an insight that EN1993-1-6 overpredicted the ultimate strength for majority of the models.

The disagreement significantly increases when the member slenderness and bolt spacing increases. It provides very unsafe predictions for members with high slenderness and bolt spacing. Those result suggested that analytical resistance calculation according to EN1993-1-6 is not applicable for this type of cross section with the predefined parameters, since the expression was derived for flexural buckling due to bending and does not take into account distortional buckling, which exists and was dominant mode in this case.

Taking the general overview of verification on ultimate resistance between the FE results and analytical calculation according to Eurocode, it can be seen that the studied columns show an in-between post-buckling behaviour of perfect column and shell structures. It should be treated as built-up member composed of plates and hence to be designed accordingly.

As for comparison, Table 15 shows the ratio of FEM-to-analytical calculation results.

Table 15. Comparison of ultimate resistance of the studied column according to FE analysis and design standards EC1993-1-3 and EC1993-1-6

(a) *b*-parameter=3

Model ID	d	t	λ	s/d	Area	EC3-1-3		EC3-1-6	FE analysis	FE/ EC3-1-3		FE/ EC3-1-6
					A_g	$N_{c,Rd}$ cross section	$N_{b,Rd}$ buckling	N_{d_shell}	$N_{FEM,Rd}$	cross section	buckling	
	[mm]	[mm]		[-]	[cm^2]	[kN]	[kN]	[kN]	[kN]	[-]	[-]	[-]
1 2 3 1 3	500	9	0.65		157.0	5003.9	4097.5	5573.4	5625.7	1.12	1.37	1.01
1 3 3 1 3	700	12			293.5	9192.4	7536.4	10418.6	10583.1	1.15	1.40	1.02
1 4 3 1 3	900	15			472.0	14653.3	12022.1	13714.0	16670.0	1.14	1.39	1.22
1 2 5 1 3	500	7			123.0	3724.3	3069.7	3495.4	4132.5	1.11	1.35	1.18
1 3 5 1 3	700	10			245.8	7425.1	6116.7	7005.2	8309.6	1.12	1.36	1.19
1 4 5 1 3	900	13			410.7	12398.7	10210.9	11721.7	13828.6	1.12	1.35	1.18
1 2 3 2 3	500	9	1	3	157.0	4927.1	3011.8	5573.4	4865.7	0.99	1.62	0.87
1 3 3 2 3	700	12			293.5	9038.6	5541.8	10418.6	8974.8	0.99	1.62	0.86
1 4 3 2 3	900	15			472.0	14391.9	8839.9	13714.0	14330.1	1.00	1.62	1.04
1 2 5 2 3	500	7			123.0	3650.2	2268.5	3495.4	3665.2	1.00	1.62	1.05
1 3 5 2 3	700	10			245.8	7213.2	4476.4	7005.2	7105.3	0.99	1.59	1.01
1 4 5 2 3	900	13			410.7	12148.3	7533.4	11721.7	12124.6	1.00	1.61	1.03
1 2 3 3 3	500	9	1.25		157.0	3020.0	1408.2	5573.4	3335.7	1.10	2.37	0.60
1 3 3 3 3	700	12			293.5	5628.8	2635.6	10418.6	6192.9	1.10	2.35	0.59
1 4 3 3 3	900	15			472.0	9024.0	4235.6	13714.0	9958.4	1.10	2.35	0.73
1 2 5 3 3	500	7			123.0	2369.0	1129.9	3495.4	2606.7	1.10	2.31	0.75
1 3 5 3 3	700	10			245.8	4710.1	2242.1	7005.2	5177.7	1.10	2.31	0.74
1 4 5 3 3	900	13			410.7	7846.1	3731.1	11721.7	8656.9	1.10	2.32	0.74

(b) *b*-parameter=4

Model ID	d	t	λ	s/d	Area	EC3-1-3		EC3-1-6	FE analysis	FE/ EC3-1-3		FE/ EC3-1-6
					A_g	$N_{c,Rd}$ cross section	$N_{b,Rd}$ buckling	N_{d_shell}	$N_{FEM,Rd}$	cross section	buckling	
	[mm]	[mm]		[-]	[cm^2]	[kN]	[kN]	[kN]	[kN]	[-]	[-]	[-]
1 2 3 1 4	500	9	0.65		157.0	4502.4	3686.9	5573.4	4982.5	1.11	1.35	0.89
1 3 3 1 4	700	12			293.5	8178.2	6704.9	10418.6	9053.7	1.11	1.35	0.87
1 4 3 1 4	900	15			472.0	12924.9	10604.0	13714.0	14571.6	1.13	1.37	1.06
1 2 5 1 4	500	7			123.0	3278.4	2702.2	3495.4	3576.7	1.09	1.32	1.02
1 3 5 1 4	700	10			245.8	6502.9	5357.0	7005.2	7358.9	1.13	1.37	1.05
1 4 5 1 4	900	13			410.7	10809.4	8902.0	11721.7	11788.0	1.09	1.32	1.01
1 2 3 2 4	500	9	1	4	157.0	4368.0	2670.0	5573.4	4555.0	1.04	1.71	0.82
1 3 3 2 4	700	12			293.5	7893.1	4839.5	10418.6	8128.1	1.03	1.68	0.78
1 4 3 2 4	900	15			472.0	12426.9	7632.9	13714.0	13158.7	1.06	1.72	0.96
1 2 5 2 4	500	7			123.0	3143.1	1953.3	3495.4	3241.9	1.03	1.66	0.93
1 3 5 2 4	700	10			245.8	6221.0	3860.7	7005.2	6373.9	1.02	1.65	0.91
1 4 5 2 4	900	13			410.7	10322.0	6400.9	11721.7	10701.6	1.04	1.67	0.91
1 2 3 3 4	500	9	1.25		157.0	4332.5	2020.3	5573.4	3269.5	0.75	1.62	0.59
1 3 3 3 4	700	12			293.5	7817.0	3660.2	10418.6	6020.5	0.77	1.64	0.58
1 4 3 3 4	900	15			472.0	12293.2	5770.1	13714.0	9777.5	0.80	1.69	0.71
1 2 5 3 4	500	7			123.0	3107.0	1481.9	3495.4	2496.8	0.80	1.68	0.71
1 3 5 3 4	700	10			245.8	6146.6	2926.0	7005.2	4985.4	0.81	1.70	0.71
1 4 5 3 4	900	13			410.7	10189.7	4845.6	11721.7	8400.3	0.82	1.73	0.72

(c) *b*-parameter=5

Model ID	d	t	λ	s/d	Area	EC3-1-3		EC3-1-6	FE analysis	FE/ EC3-1-3		FE/ EC3-1-6
					A_g	$N_{c,Rd}$ cross section	$N_{b,Rd}$ buckling	N_{d_shell}	$N_{FEM,Rd}$	cross section	buckling	
	[mm]	[mm]		[-]	[cm^2]	[kN]	[kN]	[kN]	[kN]	[-]	[-]	[-]
1 2 3 1 5	500	9	0.65		157.0	4082.8	3343.3	5573.4	4531.8	1.11	1.36	0.81
1 3 3 1 5	700	12			293.5	7332.6	6011.6	10418.6	8140.8	1.11	1.35	0.78
1 4 3 1 5	900	15			472.0	11491.3	9427.9	13714.0	12630.0	1.10	1.34	0.92
1 2 5 1 5	500	7			123.0	2912.5	2400.6	3495.4	3161.9	1.09	1.32	0.90
1 3 5 1 5	700	10			245.8	5736.4	4725.6	7005.2	6417.5	1.12	1.36	0.92
1 4 5 1 5	900	13			410.7	10415.4	8577.5	11721.7	10664.4	1.02	1.24	0.91
1 2 3 2 5	500	9	1	5	157.0	3890.3	2378.1	5573.4	4126.2	1.06	1.74	0.74
1 3 3 2 5	700	12			293.5	6917.5	4241.3	10418.6	7400.3	1.07	1.74	0.71
1 4 3 2 5	900	15			472.0	10758.1	6607.9	13714.0	11412.8	1.06	1.73	0.83
1 2 5 2 5	500	7			123.0	2720.3	1690.5	3495.4	2840.9	1.04	1.68	0.81
1 3 5 2 5	700	10			245.8	5323.7	3303.8	7005.2	5661.1	1.06	1.71	0.81
1 4 5 2 5	900	13			410.7	8752.7	5427.7	11721.7	9216.9	1.05	1.70	0.79
1 2 3 3 5	500	9	1.25		157.0	3836.6	1789.0	5573.4	3176.2	0.83	1.78	0.57
1 3 3 3 5	700	12			293.5	6798.7	3183.3	10418.6	5901.9	0.87	1.85	0.57
1 4 3 3 5	900	15			472.0	10545.4	4949.7	13714.0	9320.6	0.88	1.88	0.68
1 2 5 3 5	500	7			123.0	2664.5	1270.9	3495.4	2405.3	0.90	1.89	0.69
1 3 5 3 5	700	10			245.8	5206.3	2478.4	7005.2	4683.0	0.90	1.89	0.67
1 4 5 3 5	900	13			410.7	8538.3	4060.3	11721.7	7913.5	0.93	1.95	0.68

Based on Table 15, a relation between ultimate resistance of semi-closed polygonal profiles and the 'equivalent' cylindrical shell can be made. Figure 103 and 104 shows the ultimate resistance polygonal-cylinder ratio ($P_{u\text{-}poly}/P_{u\text{-}cshell}$) versus member global slenderness ($\bar{\lambda}$) of all columns. As shown in the subchapter 4.3 Factorial design, the most significant factor interaction are member slenderness-bolt spacing and member slenderness-diameter, hence scatter plot was made for this two factor interactions. It can be seen that resistance of semi-closed polygonal profiles are higher than the same diameter circular shell when bolt spacing ratio are 3 and for global slenderness of 0.65. The higher the bolt spacing and member slenderness, the polygonal profiles tend to have declining resistance and hence pose lower capacity than the circular ones.

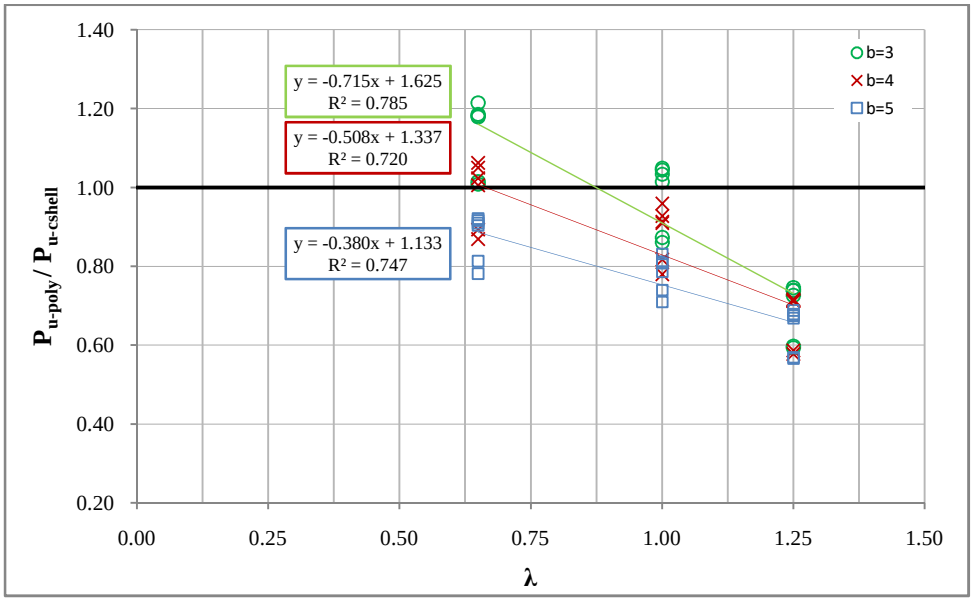


Figure 104. Comparison of resistance between polygonal vs. cylindrical shell (scatters' icon in varied *b*)

As for diameter point of view, resistance of semi-closed polygonal profiles are higher than the same diameter circular shell for diameter 900, 700 and 500, with global slenderness of 0.65 and bolt spacing ratio of 3. Similarly as above, the higher the member slenderness, the polygonal profiles tend to have declining resistance and hence pose lower capacity than the circular ones.

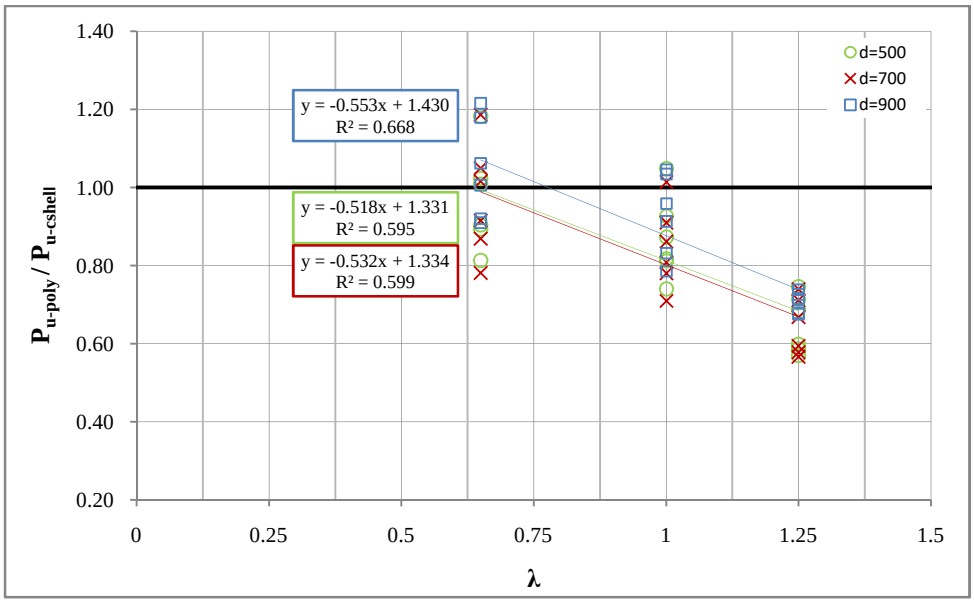
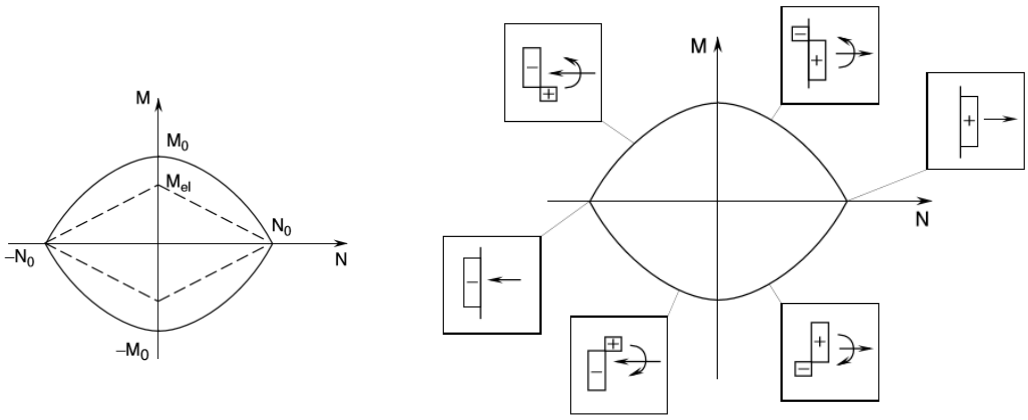


Figure 105. Comparison of resistance between polygonal vs. cylindrical shell (scatters' icon in varied *d*)

4.3.4 Axial Compression and Bending Moment

Structural members subjected to axial compression and bending moment are known as beam-column. When a doubly symmetric cross-section is subjected to axial compression and bending moment about its minor axis, the member may fail flexurally by either yielding or local buckling at the location of the maximum moment. Since in this type of profiles the shear center coincides with the center of gravity, therefore torsional-flexural mode can be neglected. For torsionally non-susceptible shapes with closed circular tubes like the ones in this study, when the end bending moment is applied, the member may fail flexurally about the x-axis or y-axis.

The resistance of a cross-section subject to compression and bending moment can be shown by the interaction curve between the normal force N_{Rd} and the internal bending moment M_{Rd}. The interaction curve has some significant points, shown in Figure 105. These points represent the stress distributions on the cross section. The internal moments and axial loads belonging to these stress distributions can be easily calculated elastically or plastically using equilibrium for the classification of cross-section.



(a) Elastic and plastic limit envelopes for a rectangular cross section **(b) Plastic limit envelope with stress distributions**

Figure 106.Interaction curve for combined bending (M) and axial force (N)

EC1993-1-3 suggests the interaction between axial force and bending moment to be obtained from second-order analysis of the member, based on the properties of the effective cross-section. An interaction formula is defined, as follows.

$$\left(\frac{N_{Ed}}{N_{b,Rd}}\right)^{0.8} + \left(\frac{N_{Ed}}{N_{b,Rd}}\right)^{0.8} \leq 1.0$$

where $N_{b,Rd}$ is the design the design buckling resistance of a compression member (flexural, torsional or torsional-flexural buckling) and $M_{b,Rd}$ is the design bending moment resistance and M_{Ed} includes the effects of shift of neutral axis, if relevant.

In this study, 4-points N-M interaction curves were created. The ultimate axial resistance (P_u) was taken for the applied bending moment with a proportion to the plastic moment resistance, namely $0.05M_u$, $0.1M_u$, and $0.15M_u$. These values of applied bending moment were taken with consideration that the column members in this study are in the form of lattice structure where ideally no bending moment exists. However, in practical condition a relatively small bending moment may occur,

induced by unbalanced axial forces from diagonals. Therefore a small proportion of plastic moment capacity was taken. Total of 162 models with combined axial compression and bending moment were created and analysed.

Axial compression load, CF3, was applied at one end of the column while bending moment about *x*-axis, CM1, applied at middle connection of the member.

Table 16.Ultimate resistance and corresponding displacement for models subject to axial compression and bending moment

(a) *d*-parameter=500

Model ID	Dia.	Thick.	Slend.	bolt spacing	Max. load	Shortening	Max. load	Shortening	Max. load	Shortening	Max. load	Shortening
	d	t	λ	s/d	N		N-M (05)		N-M (10)		N-M (15)	
					P_u	disp	P_u	disp	P_u	disp	P_u	disp
	[mm]	[mm]	[-]	[-]	[kN]	[mm]	[kN]	[mm]	[kN]	[mm]	[kN]	[mm]
1 2 3 1 3	500	9	0.65	3	5625.7	29.13	5605.36	29.03	5545.8	28.72	5466.43	28.29
1 2 3 1 4				4	4982.5	26.03	5014.15	26.34	5036.6	26.45	5043.99	26.32
1 2 3 1 5				5	4531.8	24.75	4508.34	24.49	4483.6	24.23	4465.95	24.16
1 2 3 2 3			1	3	4865.7	37.93	4666.12	36.39	4520.4	35.23	4401.06	34.38
1 2 3 2 4				4	4555.0	35.44	4690.57	36.39	4744.3	36.95	4601.61	35.84
1 2 3 2 5				5	4126.2	32.17	4135.20	32.28	4065.0	31.80	3979.55	31.08
1 2 3 3 3			1.25	3	3335.7	34.09	3363.15	33.56	3321.8	33.91	3275.82	33.48
1 2 3 3 4				4	3269.5	33.10	3306.80	32.62	3269.4	32.29	3219.05	32.11
1 2 3 3 5				5	3176.2	31.49	3231.36	31.82	3199.2	31.26	3155.90	31.12
1 2 5 1 3		7	0.65	3	4132.5	27.34	4123.04	27.26	4092.6	27.06	4058.97	26.86
1 2 5 1 4				4	3576.7	24.21	3590.68	24.29	3596.1	24.31	3595.79	24.39
1 2 5 1 5				5	3161.9	23.20	3151.29	23.02	3130.2	22.68	3082.00	21.02
1 2 5 2 3			1	3	3665.2	36.35	3868.25	38.25	3797.8	37.65	3677.00	36.57
1 2 5 2 4				4	3241.9	32.20	3268.79	32.45	3227.8	32.15	3178.39	31.61
1 2 5 2 5				5	2840.9	28.59	2803.88	28.36	2768.9	28.05	2734.77	27.83
1 2 5 3 3			1.25	3	2606.7	33.78	2682.04	32.95	2616.0	33.37	2565.73	32.58
1 2 5 3 4				4	2496.8	31.14	2565.54	31.90	2572.0	31.91	2512.11	31.39
1 2 5 3 5				5	2405.3	30.03	2473.61	30.67	2694.5	33.12	2420.80	29.99

(b) *b*-parameter=700

Model ID	Dia.	Thick.	Slend.	bolt spacing	Max. load	Shortening	Max. load	Shortening	Max. load	Shortening	Max. load	Shortening
	d	t	λ	s/d	N		N-M (05)		N-M (10)		N-M (15)	
					P_u	disp	P_u	disp	P_u	disp	P_u	disp
	[mm]	[mm]	[-]	[-]	[kN]	[mm]	[kN]	[mm]	[kN]	[mm]	[kN]	[mm]
1 3 3 1 3	700	12	0.65	3	10583.1	41.10	10451.10	40.54	10282.8	39.92	10116.05	39.30
1 3 3 1 4				4	9053.7	35.59	9129.54	35.51	9188.0	35.69	9231.05	35.87
1 3 3 1 5				5	8140.8	33.86	8166.73	34.09	8124.0	33.62	7977.14	31.25
1 3 3 2 3			1	3	8974.8	52.52	8689.70	50.90	8469.3	49.77	8272.49	48.72
1 3 3 2 4				4	8128.1	47.34	8263.92	48.24	8202.6	47.90	8033.41	47.04
1 3 3 2 5				5	7400.3	43.45	7346.71	43.11	7212.7	42.33	7094.29	41.70
1 3 3 3 3			1.25	3	6192.9	47.06	6401.78	46.04	6215.2	47.12	6088.92	45.39
1 3 3 3 4				4	6020.5	45.09	6150.01	45.43	6284.8	46.69	6052.44	45.10
1 3 3 3 5				5	5901.9	43.72	6029.10	44.63	6401.1	46.10	5909.22	43.29
1 3 5 1 3		10	0.65	3	8309.6	38.52	8190.04	37.96	8055.9	37.39	7938.30	36.86
1 3 5 1 4				4	7358.9	34.13	7341.72	34.05	7270.4	33.72	7169.81	33.33
1 3 5 1 5				5	6417.5	32.86	6408.45	32.98	6351.3	32.44	6090.13	28.48
1 3 5 2 3			1	3	7105.3	49.55	7334.15	51.22	6955.5	48.71	6813.75	47.73
1 3 5 2 4				4	6373.9	44.55	6515.84	45.36	6567.2	45.79	6461.13	45.10
1 3 5 2 5				5	5661.1	40.01	5593.41	39.46	5520.2	39.03	5455.60	38.78
1 3 5 3 3			1.25	3	5177.7	47.83	5147.44	47.26	5090.7	48.07	5037.88	46.74
1 3 5 3 4				4	4985.4	44.02	5019.61	44.96	4947.7	43.99	4872.77	43.39
1 3 5 3 5				5	4683.0	40.90	4729.89	41.38	4746.5	41.93	4691.82	41.49

(c) *d*-parameter=900

Model ID	Dia.	Thick.	Slend.	bolt spacing	Max. load	Shortening	Max. load	Shortening	Max. load	Shortening	Max. load	Shortening
	d	t	λ	s/d	N		N-M (05)		N-M (10)		N-M (15)	
					P_u	disp	P_u	disp	P_u	disp	P_u	disp
	[mm]	[mm]	[-]	[-]	[kN]	[mm]	[kN]	[mm]	[kN]	[mm]	[kN]	[mm]
1 4 3 1 3	900	15	0.65	3	16670.0	51.85	16582.64	51.52	16390.7	50.90	16126.15	50.01
1 4 3 1 4				4	14571.6	45.18	14640.38	45.50	14498.1	44.96	14491.32	45.02
1 4 3 1 5				5	12630.0	41.79	12678.90	42.07	12662.3	41.89	12435.65	40.08
1 4 3 2 3			1	3	14330.1	66.74	14622.25	68.38	14090.3	65.81	13637.78	63.91
1 4 3 2 4				4	13158.7	61.34	13515.38	62.97	13774.2	64.35	13247.31	61.89
1 4 3 2 5				5	11412.8	53.88	11446.37	54.06	11255.5	52.95	11059.90	51.96
1 4 3 3 3			1.25	3	9958.4	60.53	9967.40	60.51	9732.7	60.03	9732.66	60.03
1 4 3 3 4				4	9777.5	58.85	9806.37	59.72	9638.8	57.23	9424.57	55.67
1 4 3 3 5				5	9320.6	54.80	9345.13	54.63	9249.1	54.34	9135.92	53.75
1 4 5 1 3		13	0.65	3	13828.6	49.28	13775.28	49.20	13673.4	48.85	13559.91	48.47
1 4 5 1 4				4	11788.0	43.61	11839.70	43.74	11871.3	43.93	11869.11	43.73
1 4 5 1 5				5	10664.4	42.53	10588.47	41.91	10487.2	41.25	10408.13	40.40
1 4 5 2 3			1	3	12124.6	65.03	12608.89	67.59	12315.8	66.26	11998.24	64.66
1 4 5 2 4				4	10701.6	57.72	10806.27	57.96	10639.7	57.05	10480.09	56.24
1 4 5 2 5				5	9216.9	50.32	9074.74	49.42	8937.6	48.51	8813.19	48.05
1 4 5 3 3			1.25	3	8656.9	61.02	8678.68	60.56	8581.1	60.92	8464.29	60.53
1 4 5 3 4				4	8400.3	57.10	8399.80	56.50	8293.8	56.23	8188.90	55.65
1 4 5 3 5				5	7913.5	53.21	7874.48	52.71	7773.0	52.03	7669.76	51.49

Table 16 shows the results of ultimate resistance and corresponding displacement for models subject to axial compression and bending moment. As expected, the presence of bending moment slightly lowers the ultimate strength of the columns due to the interaction between them. In average, the ultimate load decreased 0.52% by applying bending moment. However, some models show an anomaly where the ultimate strength went higher as bending moment was applied. A close examination on the failure state of the members suggested that the increase happened due to the favourable effect of moment direction on the flexural mode of the column under axial loading. It gives stabilizing effect to the secondary moment resulting from the applied axial load *P* and the deflection of the member. Then at a point of certain increased applied moment, the flexural mode changes direction and the resistance decent. The application of bending moment will change the contribution from other modes, i.e. flexural mode on the interaction and hence change the failure mode.

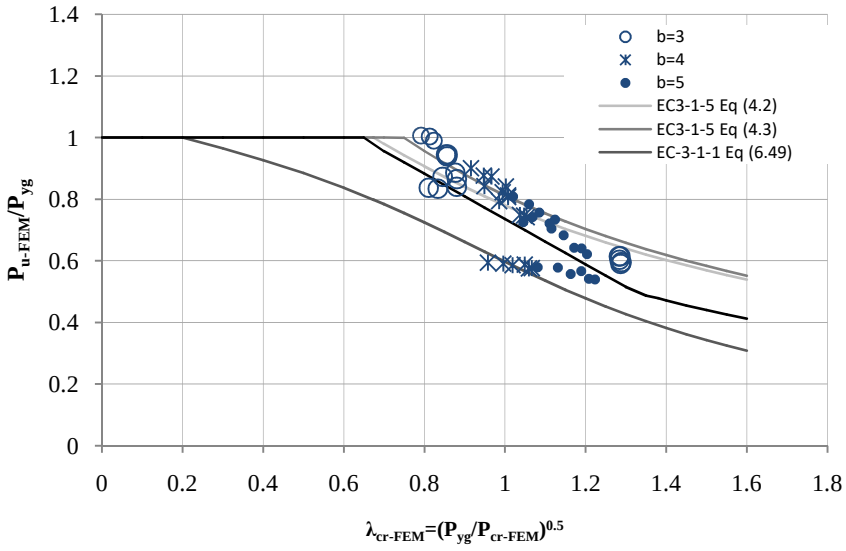


Figure 107.Parametric study results of the studied columns (normalized resistance based on P_{yg} vs. slenderness based on P_{yg} and FEM critical buckling loads): Models RIKS-NM05

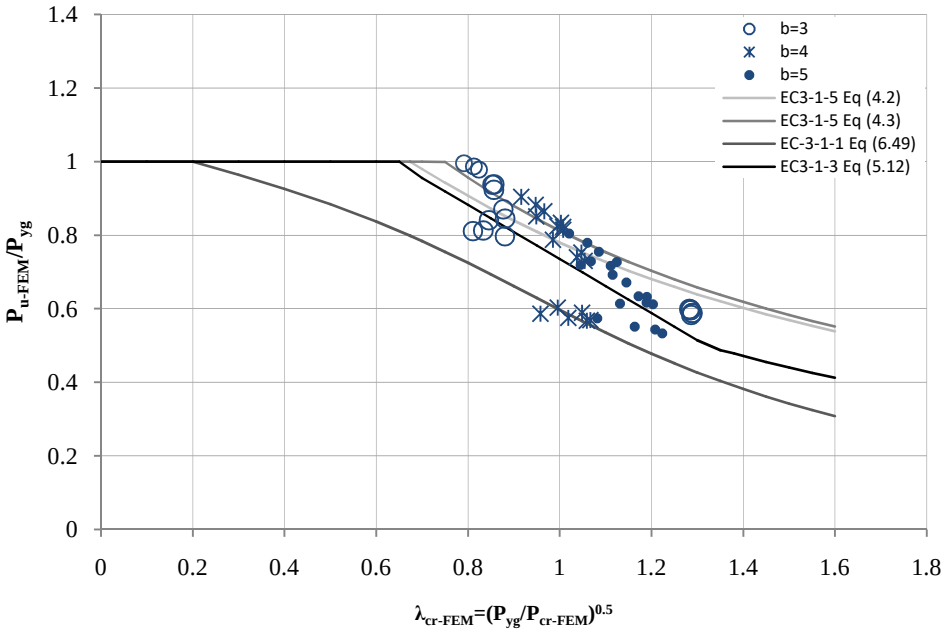


Figure 108.Parametric study results of the studied columns (normalized resistance based on P_{yg} vs. slenderness based on P_{yg} and FEM critical buckling loads): Models RIKS-NM10

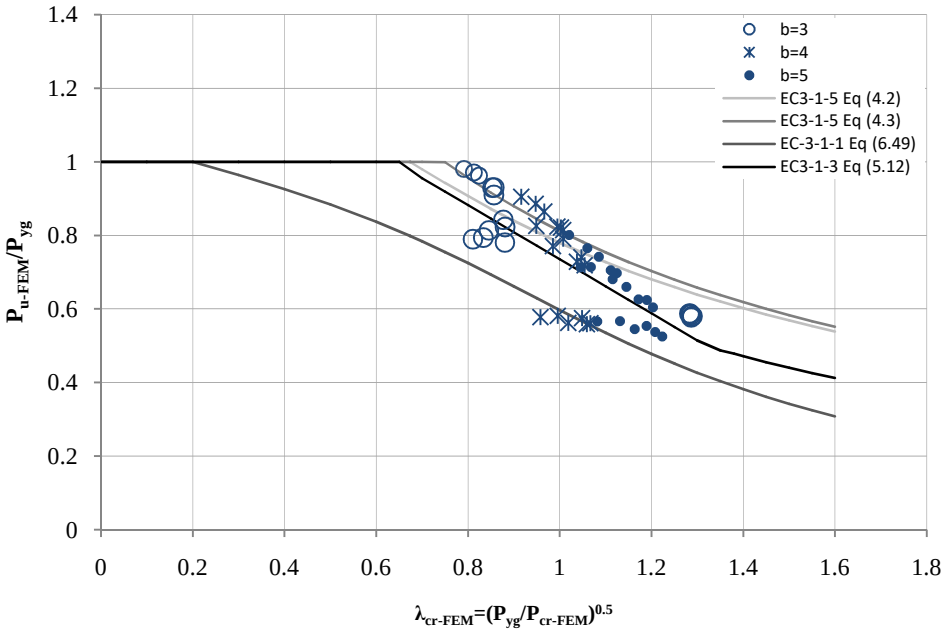


Figure 109.Parametric study results of the studied columns (normalized resistance based on P_{yg} vs. slenderness based on P_{yg} and FEM critical buckling loads): Models RIKS-NM15

Figure 106 – 108 shows the normalized resistance versus slenderness of the models subject to combined axial compression and bending moment. It can be seen that in general more scatter of data points are in unsafe region found in the models. Models with low slenderness tend to be more affected by the moment application, while some models with high slenderness were more likely to experience an increase of ultimate strength.

Axial – bending moment interaction curves were constructed to give a clearer picture of the influence of N-M interaction on the ultimate strength of the models. The curve compared group of models with the same diameter and cross-section slenderness, different lambda and bolt spacing.

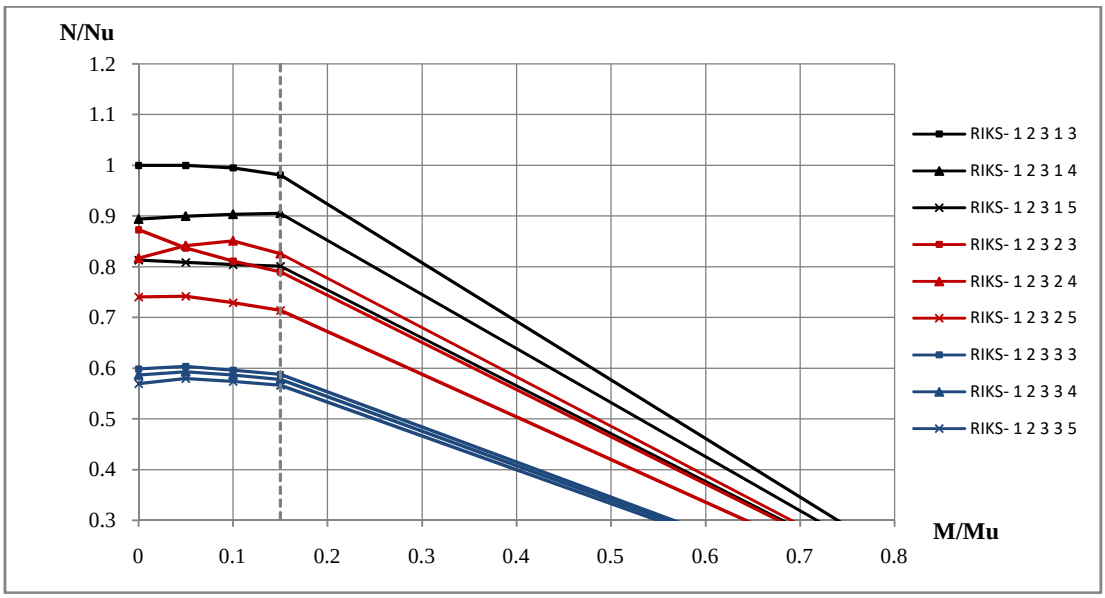


Figure 110. N-M interaction curve for models with kept dia. 500 and slend. 90; varied lambda and bolt spacing

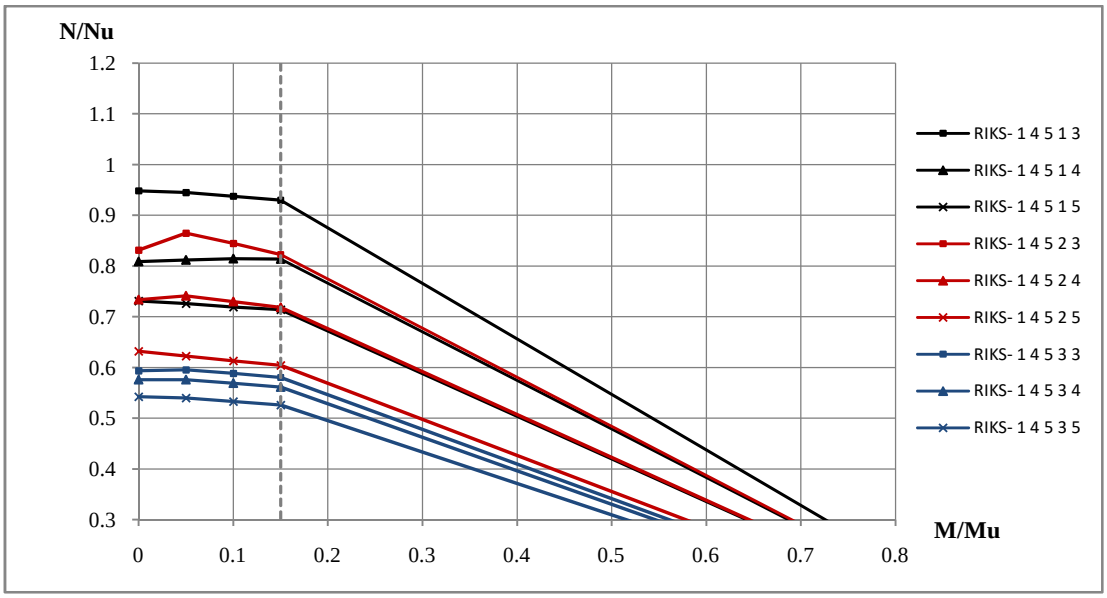


Figure 111. N-M interaction curve for models with kept dia. 900 and slend. 110; varied lambda and bolt spacing

The axial-bending moment interaction curve shows that the parameters did influence significantly on the effect of bending moment to the ultimate strength of the members. Slenderness λ-parameter seemed to give more considerably effect on the N-M interaction, compared to other parameters. The effect of interaction was higher with higher slenderness. More slender columns were also more likely to have higher contribution from flexural mode in the interaction of failure. Complete N-M interaction curve can be seen in Annex B.3.

Figure 111 shows failure mode of model 1-2-5-3-5 under pure axial compression force and combined bending with 0.05M, 0.1M, and 0,15M. By applying bending moment with various magnitudes, the failure mode shape of the member was changed and the resistance changed accordingly.

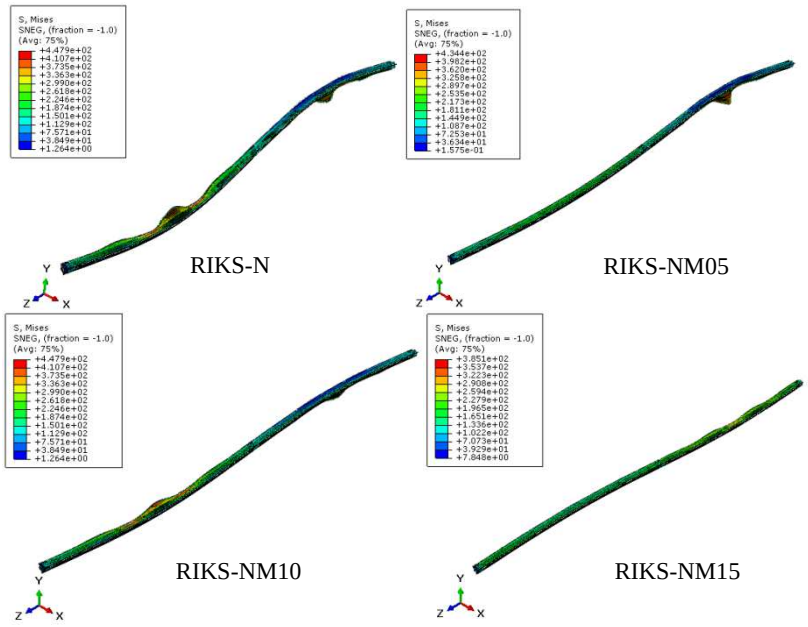


Figure 112. Failure mode of model RIKS-1-2-5-3-5

Figure 112 shows the history of rotation u1 for model 1-2-5-3-5. This affirms the change of flexural mode shape on the model due to applied bending moment. The complete data of rotation u1 of the members can be seen in Annex B.2.

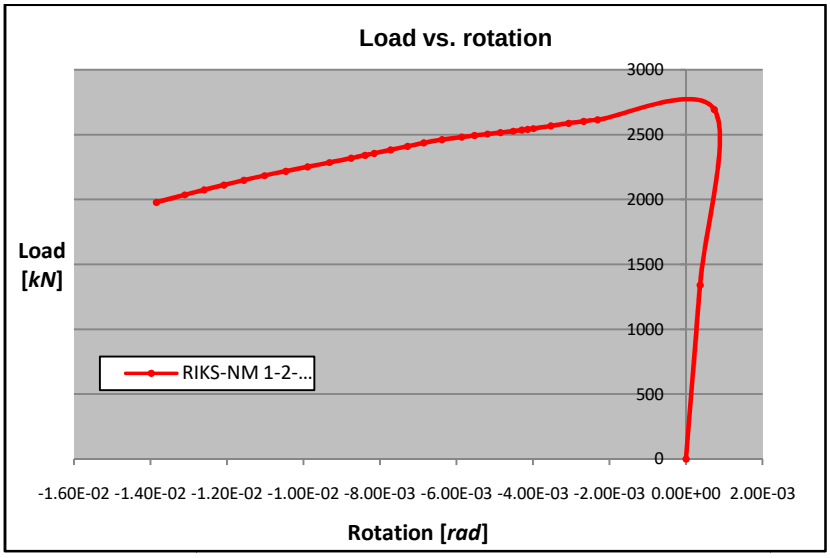


Figure 113. Rotation u1 of model RIKS-NM-1-2-5-3-5

4.3.5 Resistance-to-weight ratio

In order to assess the efficiency of a structure - normally associated with the amount of material used, a strength-to-weight ratio was calculated. It compares the weight of the structure itself to the amount of weight it can carry/support without collapsing. A very high strength-to-weight ratio can be

achieved either by optimizing the material or the structural system itself. As material, steel has a good strength-to-weight ratio; steel construction requires less material than the traditional construction technologies and contributes to reducing a building's environmental impact. Since one of the purposes of developing this type of structural member is as structural optimization, this parameter becomes important to be considered.

Due to large number of models, only samples were taken for the comparison here. Complete calculation data of s/w ratio can be seen in the Annex D. Strength-to-weight ratio was calculated for different diameter, thickness, bolt spacing and length.

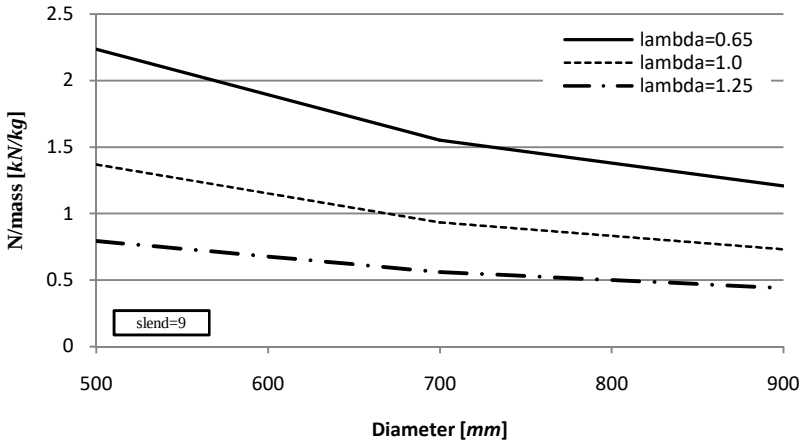


Figure 114.strength-to-weight ratio versus diameters

Profiles with diameter 500*mm* give the highest strength-to-weight ratio and with increasing diameter, the ratio decreases. The Effect is less in the members with high slenderness.

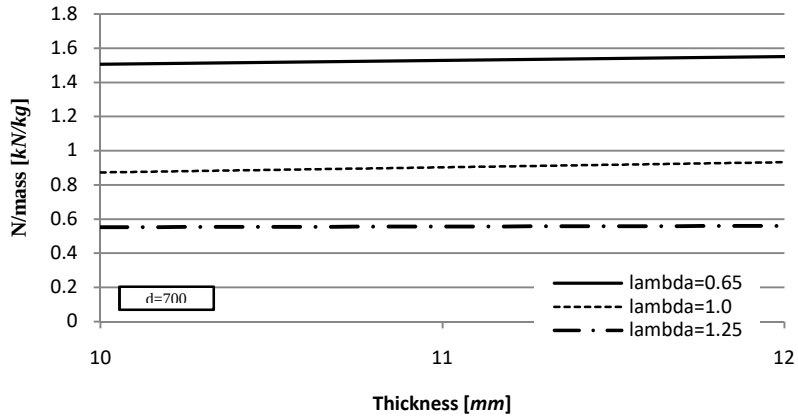


Figure 115.strength-to-weight ratio versus thicknesses

Influence of thickness is not significant on the strength-to-weight ratio. Higher thickness gives higher s/w ratio. Meanwhile, bolt spacing ratio equal to *3d* gives the highest s/w ratio. By increasing bolt spacing, the ratio will decrease, however the influence is less for high slenderness.

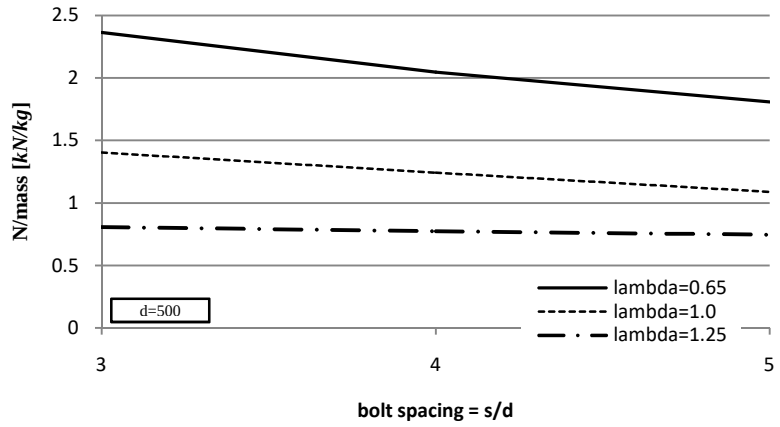


Figure 116.strength-to-weight ratio versus bolt spacing

Figure 116 shows that the influence of member slenderness on s/w ratio was significant. The ratio dropped considerably with increasing slenderness; the s/w ratio less than 1.0 was shown by members with slenderness $\bar{\lambda}$>1.

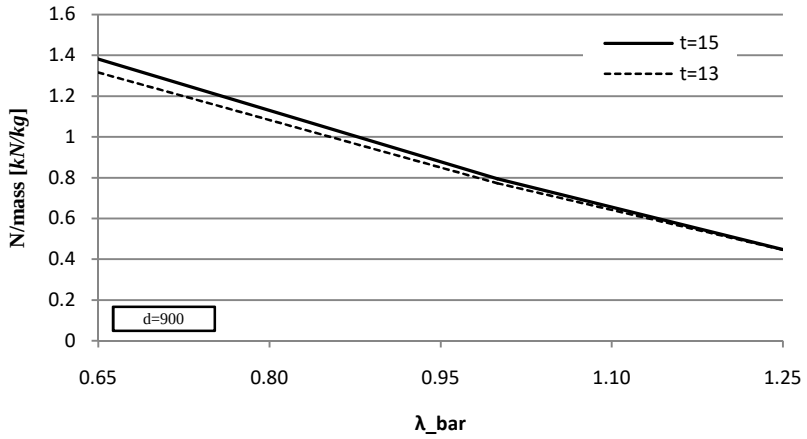


Figure 117.strength-to-weight ratio versus member slenderness

From the comparison, it can be seen that the significant parameters on the strength-to-weight ratio were member slenderness and diameter. The choice of geometrical properties in the design of semi-closed polygonal cross-section is of important aspect and will determine the resistance and effectiveness of the structure.

As a more accurate and reliable method of analysis, interaction plot from factorial design was created to see the significant factor and interaction between factors for response variable: strength-to-weight ratio. From Figure 117, it can be seen that the most significant factor for s/w ratio is member slenderness and then diameter. Meanwhile, the most significant interaction is member slenderness-diameter and member slenderness-bolt spacing.

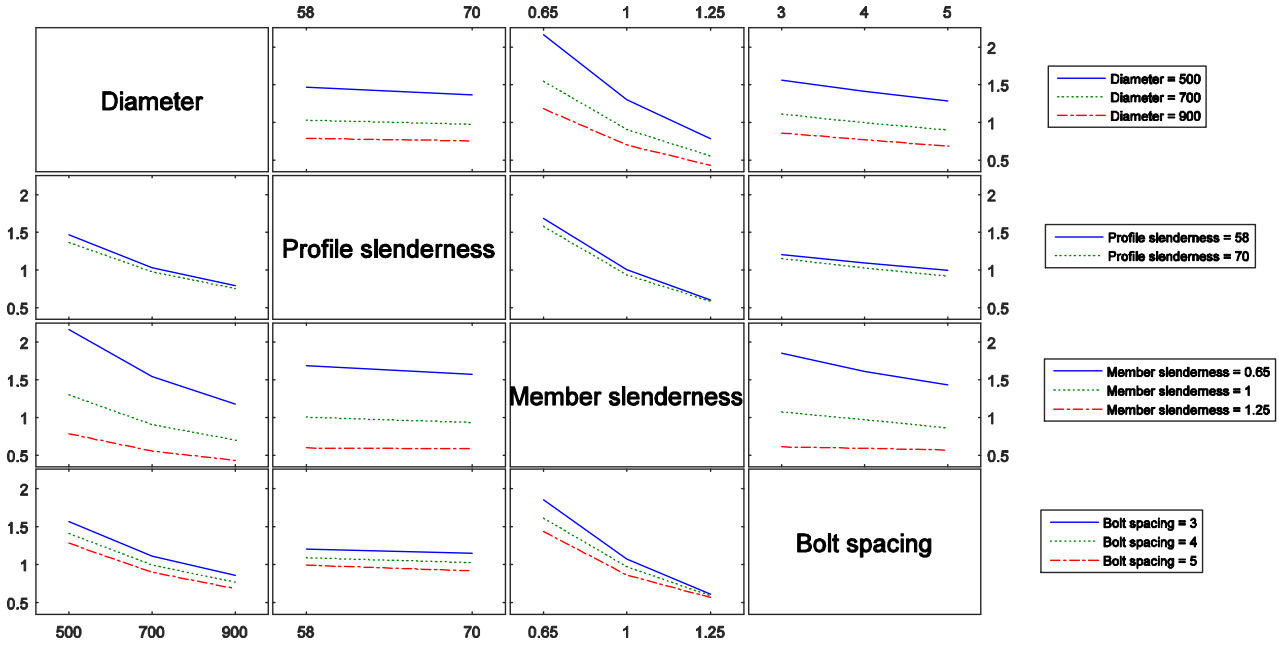


Figure 118. Interaction plot for strength-to-weight ratio

This far, from the analysis it is known that two factors in this parametric study were the most significant factor for the resistance of the members: diameter and member slenderness. As for interaction between factors, the significant interaction is from member slenderness-diameter and member slenderness bolt spacing.

5 FINITE ELEMENT MODELLING OF THE STIFFNESS ON LIPS' BOLTED CONNECTIONS

The value of connection stiffness on the lips of folded plates becomes one important parameter in the study. These stiffnesses will determine the buckling behaviour of the member as a semi-closed section and are useful for application in finite strip numerical modeling, e.g. CUFSM. For such thin walled open section, stability i.e. distortional buckling, is one of the most dangerous failure cases, it leads to brittle failure. At the end, the acquired stiffness of connection can also be used to establish effective bolting density for this type of semi-closed cross section.

5.1 Modelling and Analysis Method

As described in the introduction chapter that as individual plate, the section has very small torsional stiffness, which in turn inherit very low stability, so that it is unfavorable for compression members. It will crush at a low level of compressive loading. One way of improving the resistance is to assemble them and make the cross-section closed by using mechanical fasteners. It is called semi-closed cross-section because it is not continuously and rigidly connected. Therefore, the level of connection, in this case is its stiffness, play a vital role in the stability and resistance of the member.

Bolts were devices used to connect the folded plates to form the assembled closed section. The bolts were placed in correspondence to the middle line of each lip of folded plates longitudinally along the span. Spacing of the bolt was one parameter in the parametric study with the calculation and result already described in Chapter 3 and Chapter 4, respectively. In the modeling process, rigid-body constraint which working between paired holes perimeter represent the bolt connection.

In this chapter, analyses were carried out in FEM ABAQUS for certain models in order to obtain translational and rotational spring stiffness of the connection. The relationship between developed reaction force and relative displacement was adopted to estimate the stiffness value.

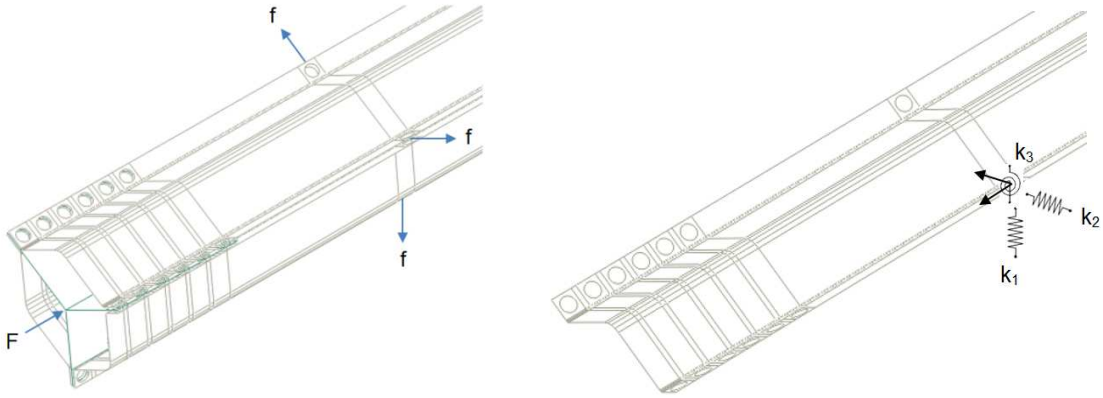


Figure 119. Concept of stiffness of the connection of folded plates assembly

In order to simulate the stiffness calculation of bolt, unit displacement is given for each bolt set in radial direction. This scheme was modeled by applying the displacement on this RP. Reaction force at the RP of the bolt set was then measured to get the spring stiffness in this direction (Figure 119).

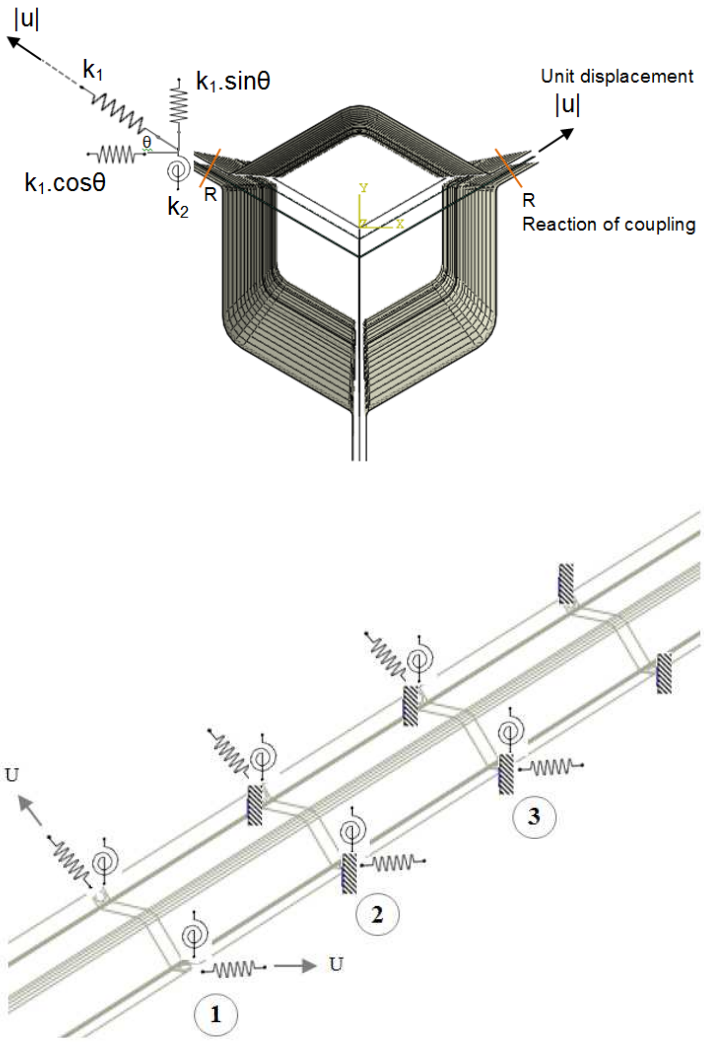


Figure 120. Schematic procedure for stiffness calculation

Steps of stiffness calculation can be summarized as follows:

1. Consider one of the three sectors. It is going to see the stiffness that the other two sectors provide
2. Looking at set of bolts independently. By "set of bolts" it means the 2 bolts of a cross-section. So, for one sector there are 2 bolts holding on every bolted cross-section
3. Construct a local stiffness matrix for one set of bolts (the 2 bolts connecting one sector to the other two)
   a. When calculating a stiffness matrix, one locks/fixes all the DOFs and activates/applies displacement on one DOF each time, measuring the reactions on all the DOFs
   b. There are a number of bolt sets on each span. Each bolt set is one DOF, namely DOF *i*. Supposing that it looks for a 3x3 stiffness matrix [K]. So, if focus given on one bolt set (one DOF), the 2 neighboring sets are the two extra DOFs of the matrix [K]. Consider as neighboring sets only from the one side, not two symmetrical sets.
   c. Lock U1, U2 for all the bolts connecting the two lips of one sector, except the two that is investigated (pick the bolt set closest to the middle of the span. This is DOF1). Apply the BCs on the RPs that are coupled to the shell. Leave the rest of the bolts (the ones connecting the other two sectors) free

d. Apply displacement on the DOF1. Both bolts take displacement on the x and y as cosine and sine function accordingly

e. Request output for the reaction forces on the RPs corresponding to the 3 DOFs (in order to request outputs for reaction forces it needs to designate sets in ABAQUS so as to create 6 sets for the 6 bolts of the 3 DOFs)

f. So, for three sequential bolt sets, DOF1 is the first, DOF2 and DOF3 are the two following bolt sets accordingly. Do the same for the next bolt set, one space next to the first set. Lock the first set and apply the displacement on the second. The same goes for the third

g. Those reaction forces will give the diagonally symmetric 3x3 [K] matrix

4. To take into account rotational stiffness:

   a. Restrain the rotations on the RPs as well. Then it will add the rotational DOF in the matrix which will become 6x6

   b. First lock rotations and apply the displacements and then lock the displacements and apply a rotation around z. In both cases read the output for the forces U1, U2 and moment UR3

Therefore, the main stiffness of each bolt is the reaction of this bolt when displacement is applied. There is some additional stiffness coming from the rest of the bolts but it will be small. These are the off-diagonal numbers on 6x6 stiffness matrix.

It can be noted that actually it only needs to apply the displacement on one bolt set (a pair of bolts) and measure the reactions on this one and the two neighboring sets. This means no repetition to be performed for the other two sets. The reactions would be the same.

$$\mathbf{K} = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix}$$

$$k_{22} = k_{33} = k_{11}$$

$$k_{21} = k_{23} = k_{32} = k_{12}$$

$$k_{31} = k_{13}$$

This also applies for moment rotation. The 6x6 [K] matrix then can be simplified as follows.

$$\begin{array}{c} \begin{matrix} u_1 & u_2 & u_3 & ur_1 & ur_2 & ur_3 \end{matrix} \\ \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} & k_{15} & k_{16} \\ k_{12} & k_{11} & k_{12} & k_{15} & k_{14} & k_{15} \\ k_{13} & k_{12} & k_{11} & k_{16} & k_{15} & k_{14} \\ k_{41} & k_{42} & k_{43} & k_{44} & k_{45} & k_{46} \\ k_{42} & k_{41} & k_{42} & k_{45} & k_{44} & k_{45} \\ k_{43} & k_{42} & k_{41} & k_{46} & k_{45} & k_{44} \end{bmatrix} \end{array}$$

The shaded quadrants which come from rotational reaction due to translational displacement and translational reaction due to rotational displacement, respectively, can be neglected due to small

values of them. The main diagonal as the most significant entities is of interest in this study, i.e. $\mathbf{k_{11}}$ and $\mathbf{k_{44}}$. These two values are going to be used in CUFSM as translational stiffness and rotational stiffness, respectively, of the interaction between folded plates.

In finite element strip software package CUFSM, there is no tool for applying point-based connection to assemble thin-walled profiles since the software is mostly used for open cross-section. Alternatively, stiffness per unit length can be applied between profiles so that the profiles will be considered as a semi-closed cross-section. Therefore, FEM modelling was carried out in order to get proper stiffness values to be used in CUFSM. Compared to ABAQUS, CUFSM has peculiar ability in providing buckling stress of each buckling mode, interaction of them and contribution from each mode so that a clear picture of buckling behaviour of the thin-walled member can be given.

Some models in the parametric studies were used for calculation of lips' connection stiffness. Modelling was done by modifying the Python script for intended parameters used in the calculation. Parameters such as number of corners, diameter, and cross-section slenderness were modified on the loop lines. Example of script for model with *n*=12, *d*=700, *slend*=140 (thickness=8*mm*) is shown below.

```
for i in range(2,3,1):
 for j in range(2,3,1):
       for k in range(7,8,1):
             for l in range(1):
```

As for bolt spacing, the *s/d* ratio can be changed to define the space value.

```
 # Loop through the different bolt spacings (temporary b to change)
 b = [3, 4, 5]
```

Static general analysis was used in this analysis, therefore the script should be modified as follows.

```
load_step = c_model.StaticStep(
 name='Load',
 previous='Initial')
```

Sets of points were created at the reference points of rigid body constraint to accommodate history output request. Reaction forces at corresponding RPs were then measured in sequential manner as described above.

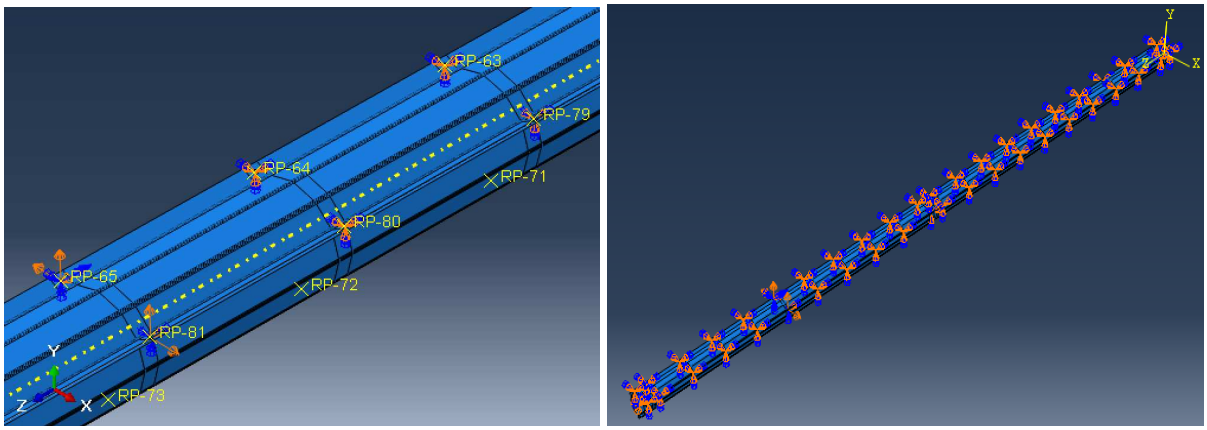


Figure 121. Application unit of boundary condition and unit displacement

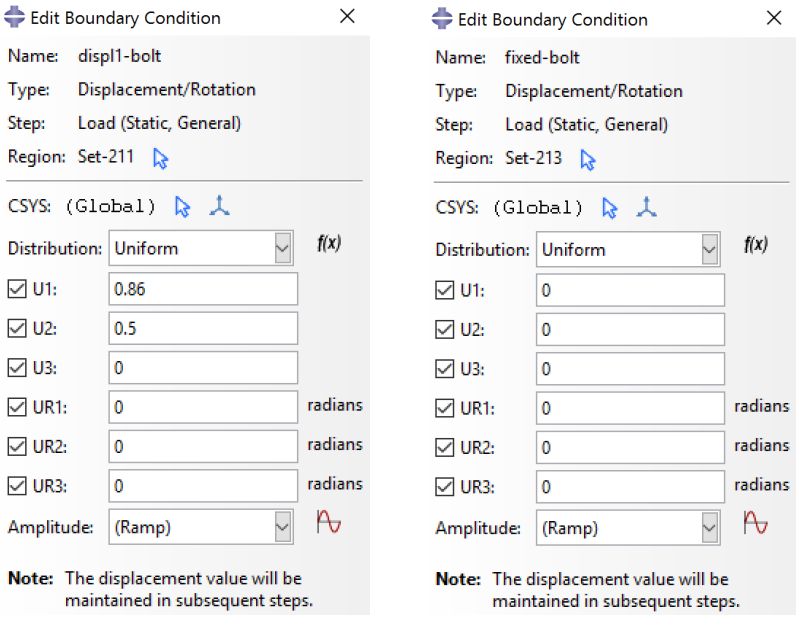


Figure 122. Set of boundary conditions at active DOF and locked DOF

5.2 Results

The main purpose of this analysis is to get spring values to be fed to finite strip package CUFSM which is used in another thesis (Jimmy and Hamse, 2017). The supplementary functionalities in CUFSM can be used to compare and complement the results obtained in this thesis. Another purpose is to see the influence of some parameters to the stiffness value.

5.2.1 Influence of diameter and plate thickness on the stiffness of lips' connection

Results for translational spring value due to translational (u1, u2, u3) and rotational (ur1, ur2, ur3) load are shown in Table 17. It can be seen that translational stiffness of the consecutive bolts has the most significant value on the first bolt ($\mathbf{k_{11}}$) where the load is applied, and reduces significantly on the third bolt ($\mathbf{k_{13}}$) with negligible value. It is the reason that in the analysis three bolts were taken and measured in sequential manner. This way, the connection stiffness that the other two sectors provide can be obtained while taking into account the neighboring bolts.

Translational stiffnesses due to rotational load, indicated as $\mathbf{k_{14}}$, $\mathbf{k_{15}}$, and $\mathbf{k_{16}}$ in the matrix, have small values and can be neglegted.

Results for rotational spring due to translational (u1, u2, u3) and rotational (ur1, ur2, ur3) load are shown as well in Table 117. The same principle as described above applies to the rotational stiffness, the first bolt ($\mathbf{k_{44}}$) due to rotational load has the most significant value while the remaining reduces as distance increases.

Likewise, rotational stiffnesses due to translational load, indicated as $\mathbf{k_{41}}$, $\mathbf{k_{42}}$, and $\mathbf{k_{43}}$ in the matrix, have small values and can be neglegted.

Table 17. Stiffness of lip connection for models with different diameters (*d*) and thicknesses (*t*)

Model				u1	u2	u3	ur1	ur2	ur3
				transl. spring (kN)					
n	b	d	t	**k11**	k12	k13	k14	k15	k16
1	2	300	4	**58.860**	28.139	2.237	1.369	0.854	0.119
1	2	300	5	**84.258**	38.564	3.292	1.681	1.080	0.164
1	2	300	6	**114.344**	49.901	4.490	1.913	1.276	0.204
1	2	500	5	**43.513**	18.722	2.029	1.506	0.831	0.119
1	2	500	6	**61.585**	26.500	2.931	1.913	1.276	0.204
1	2	500	8	**105.791**	44.482	5.117	2.803	1.737	0.152
1	2	500	11	**193.493**	76.391	9.015	3.583	2.503	0.332
1	2	700	8	**68.489**	27.898	3.775	2.723	1.504	0.077
1	2	700	11	**129.080**	51.878	6.505	4.243	2.581	0.140
1	2	700	15	**239.306**	91.565	11.558	5.361	3.778	0.438
1	2	900	11	**94.022**	36.679	5.700	4.292	2.365	0.145
1	2	900	15	**173.782**	66.888	9.192	6.194	3.847	0.185
				rot. spring (kNm)					
				k41	k42	k43	**k44**	k45	k46
1	2	300	4	0.110	0.030	0.010	**18.652**	6.801	1.013
1	2	300	5	0.169	0.040	0.013	**28.367**	8.439	1.387
1	2	300	6	0.243	0.047	0.015	**41.108**	9.781	1.705
1	2	500	5	0.079	0.007	0.017	**30.301**	10.042	0.045
1	2	500	6	0.243	0.047	0.015	**44.792**	13.664	0.625
1	2	500	8	0.256	0.028	0.031	**82.921**	19.905	2.032
1	2	500	11	0.554	0.021	0.042	**173.524**	27.658	4.118
1	2	700	8	0.153	0.002	0.031	**88.187**	23.459	0.107
1	2	700	11	0.365	0.007	0.051	**184.302**	38.953	2.782
1	2	700	15	0.857	0.011	0.064	**388.342**	54.697	7.292
1	2	900	11	0.252	0.039	0.052	**195.265**	45.031	0.566
1	2	900	15	0.528	0.073	0.080	**402.167**	70.244	4.816

Figure 122 shows that stiffnesses decrease with increasing diameter. This corresponds to the Hooke's law that length of member is inversely proportional to its stiffness. In average, with increasing diameter by 45%, the translational stiffness will experience decrease by 36%. However, effect of diameter is not significant for rotational stiffness and shows the opposite trend to the translational stiffness. In average, with increasing diameter by 45%, the rotational stiffness will experience increase by 6%.

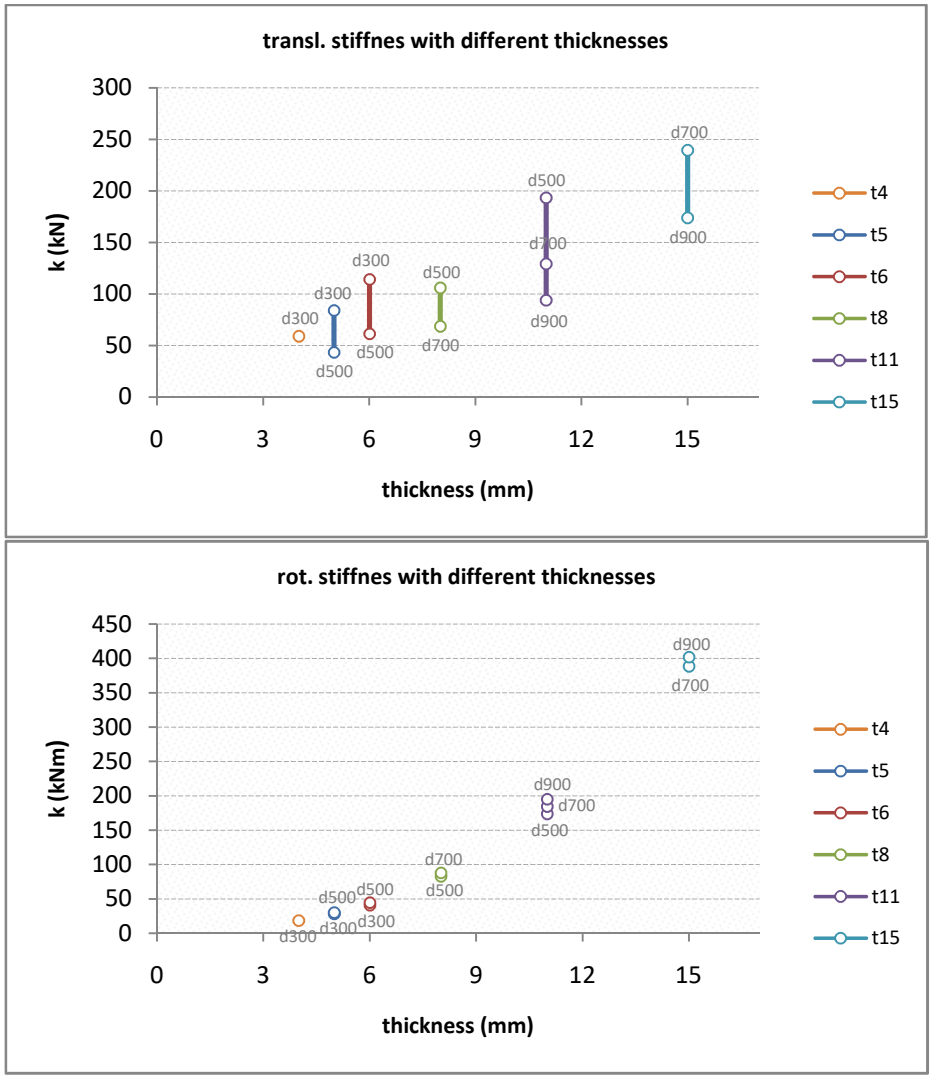


Figure 123. Stiffness with different thicknesses

d		t	diff	
			k11	k44
300		5		
500	+66.7%	5	-48.36%	+6.82%
300		6		
500	+66.7%	6	-46.14%	+8.96%
500		8		
700	+40.0%	8	-35.26%	+6.35%
500		11		
700	+40.0%	11	-33.29%	+6.21%
900	+28.6%	11	-27.16%	+5.95%
700		15		
900	+28.6%	15	-27.38%	+3.56%
avg	+45.08%		-36.26%	+6.31%

In contrary, Figure 123 shows that stiffnesses increase with increasing thickness. This also corresponds to the Hooke's law that area of cross section is proportional to its stiffness. In average, with increasing thickness by 67%, the translational stiffness will experience increase by 67%. Likewise, with increasing thickness by 67%, the rotational stiffness will experience increase by 83%.

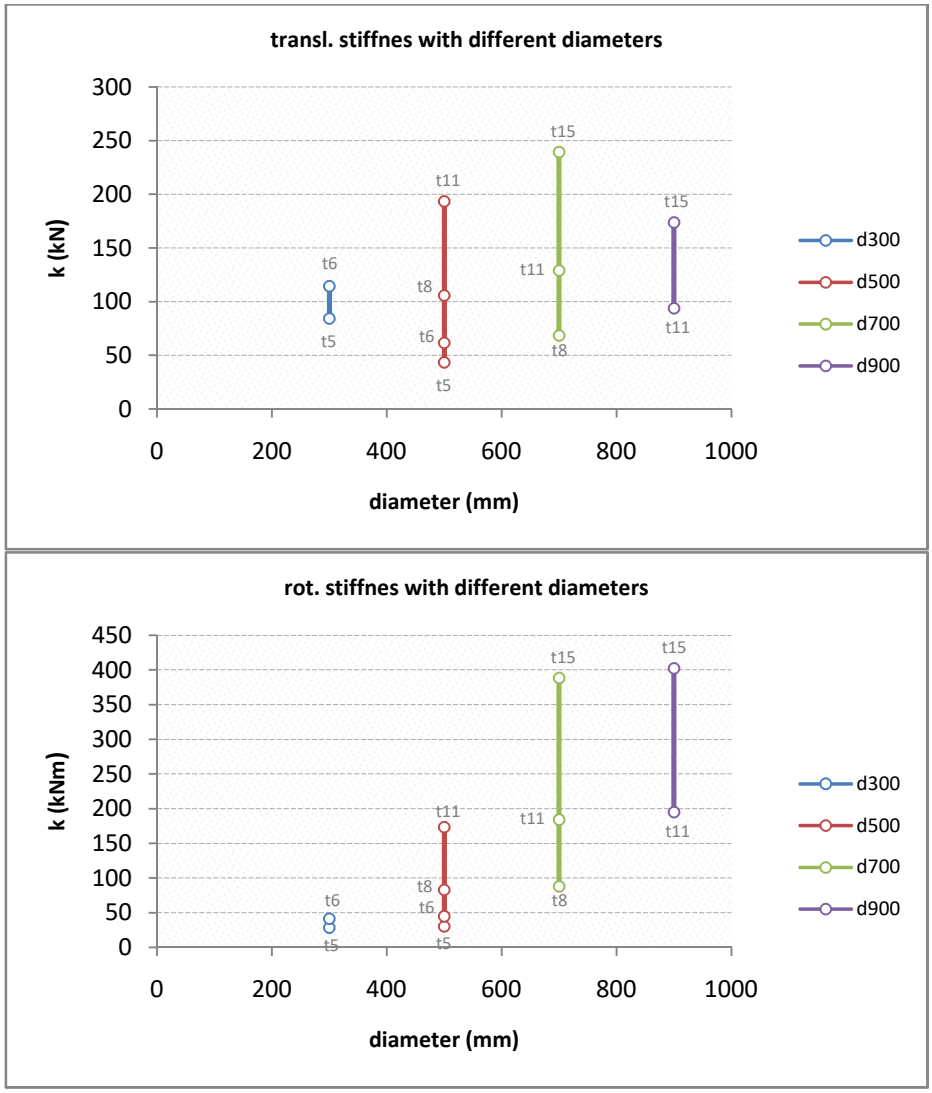


Figure 124. Stiffness with different diameter

d	t		diff	
			k11	k44
300	4			
300	5	+25.0%	+43.15%	+52.08%
300	6	+20.0%	+35.71%	+44.91%
500	5			
500	6	+20.0%	+41.53%	+47.82%
500	8	+33.3%	+71.78%	+85.12%
500	11	+37.5%	+82.90%	+109.26%
d	t		diff	
			k11	k44
700	8			
700	11	+37.5%	+88.47%	+108.99%
700	15	+36.4%	+85.39%	+110.71%
900	11			
900	15	+36.4%	+84.83%	+105.96%
avg		+66.7%	+66.72%	+83.11%

5.2.2 Influence of bolt spacing on the stiffness of lips' connection

Results of spring stiffness for different bolt spacing is presented in Table 18-19 and Figure 124. As expected, a more densely bolt connection gives higher stiffness both translation and rotation. Three variables were used, i.e. space-to-diameter ratio (*b*) equal to 1, 2, and 3, with two different diameters, i.e. 300 and 500mm and thicknesses, i.e. 5 and 6mm.

As can be seen in Figure 124, bolt spacing has significant effect on the spring value when it changed from 1*d* to 2*d*. By changing the spacing doubly, from *1d* into *2d*, the translational stiffness value reduces about 60%, and reduces about 40% with changing spacing from *2d* into *3d*. Effect of changing diameter and thickness is the same as described in the previous section. As for rotational stiffness, no significant effects from different diameters.

Table 18. Stiffness of lip connection for models d=300mm with different bolt spacing (*b*) and thicknesses (*t*)

Model				u1	u2	u3	ur1	ur2	ur3
				transl. spring (kN)					
n	b	d	t	**k11**	k12	k13	k14	k15	k16
1	1	300	5	**215.403**	97.923	9.334	4.812	2.561	0.134
1	2	300	5	**84.258**	38.564	3.292	1.681	1.080	0.164
1	3	300	5	**48.250**	19.213	1.518	0.657	0.454	0.080
1	1	300	6	**283.797**	128.262	11.052	5.867	3.231	0.118
1	2	300	6	**114.344**	49.901	4.490	1.913	1.276	0.204
1	3	300	6	**68.495**	25.379	1.819	0.726	0.524	0.092
				rot. spring (kNm)					
				k41	k42	k43	**k44**	k45	k46
1	1	300	5	0.169	0.002	0.038	**53.239**	20.607	0.824
1	2	300	5	0.169	0.040	0.013	**28.367**	8.439	1.387
1	3	300	5	0.126	0.032	0.006	**20.444**	3.453	0.698
1	1	300	6	0.239	0.004	0.044	**72.039**	25.769	0.334
1	2	300	6	0.243	0.047	0.015	**41.108**	9.781	1.705
1	3	300	6	0.189	0.028	0.013	**33.213**	7.447	1.133

Table 19. Stiffness of lip connection for models d=500mm with different bolt spacing (*b*) and thicknesses (*t*)

Model				u1	u2	u3	ur1	ur2	ur3
				transl. spring (kN)					
n	b	d	t	**k11**	k12	k13	k14	k15	k16
1	1	500	5	**101.174**	43.251	4.652	2.643	1.057	0.340
1	2	500	5	**43.513**	18.722	2.029	1.506	0.831	0.119
1	3	500	5	**26.889**	11.240	1.266	0.815	0.495	0.055
1	1	500	6	**139.461**	59.101	7.292	3.769	1.608	0.432
1	2	500	6	**61.585**	26.500	2.931	1.913	1.276	0.204
1	3	500	6	**38.054**	15.516	1.743	1.010	0.636	0.085
				rot. spring (kNm)					
				k41	k42	k43	**k44**	k45	k46
1	1	500	5	0.054	0.018	0.009	**44.350**	13.275	4.002
1	2	500	5	0.079	0.007	0.017	**30.301**	10.042	0.045
1	3	500	5	0.075	0.019	0.010	**22.022**	5.907	0.747
1	1	500	6	0.090	0.025	0.016	**66.461**	20.009	4.967
1	2	500	6	0.243	0.047	0.015	**44.792**	13.664	0.625
1	3	500	6	0.114	0.028	0.013	**33.213**	7.447	1.133

d	t	s/d	diff		d	t	s/d	diff	
			k11	k44				k11	k44
300	5	1			500	5	1		
300	5	2	-60.88%	-46.72%	500	5	2	-56.99%	-31.68%
300	5	3	-42.74%	-27.93%	500	5	3	-38.21%	-27.32%
300	6	1			500	6	1		
300	6	2	-59.71%	-42.94%	500	6	2	-55.84%	-32.60%
300	6	3	-40.10%	-19.20%	500	6	3	-38.21%	-25.85%
avg			-50.86%	-34.20%	avg			-47.31%	-29.36%

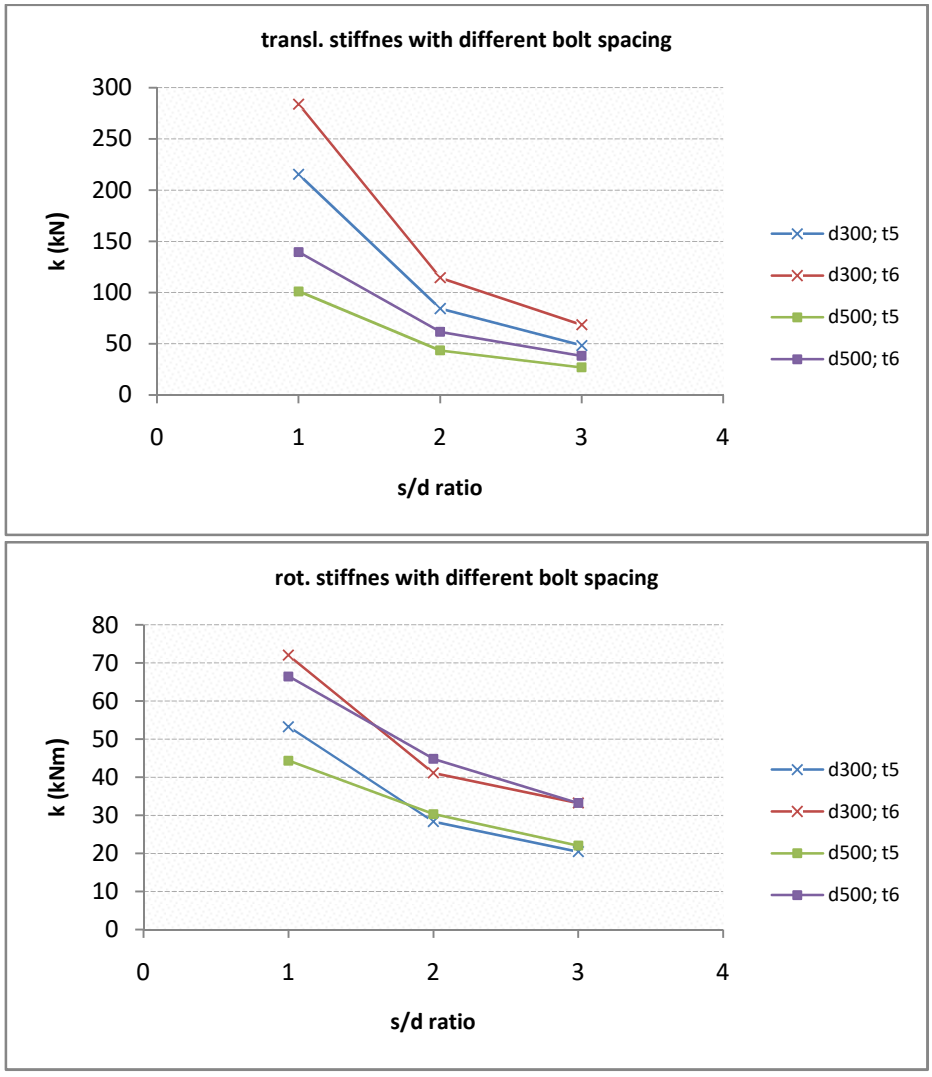


Figure 125. Stiffness with different bolt spacing

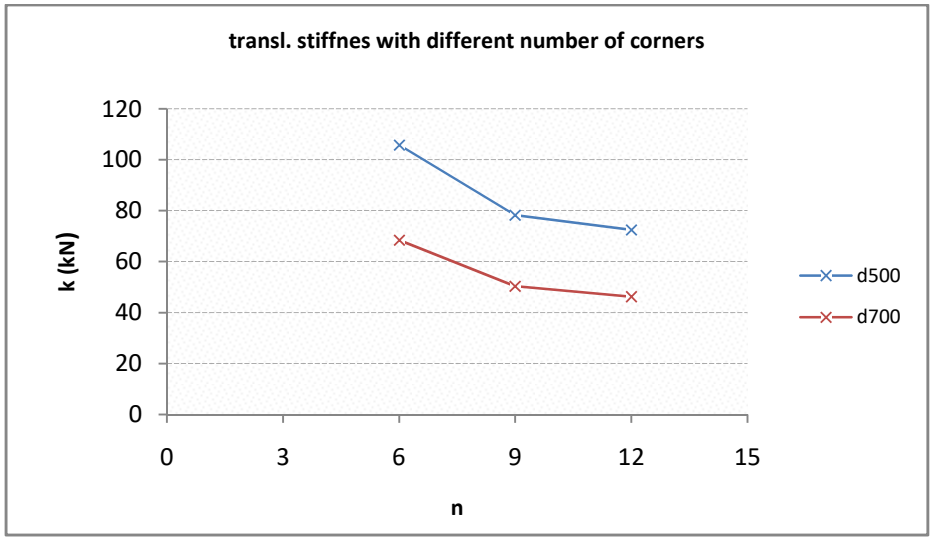
5.2.3 Influence of number of corners on the stiffness of lips' connection

Table 20 and Figure 125 show that different number of corners does not have significant effect on the spring values. Smaller number of corners gives higher stiffness since smaller angleof bend provideshigher stiffness to the member.Effect on rotational stiffnes is not significant compared to the translational one.

Table 20. Stiffness of lip connection for models with different number of corners (*n*) and diameter (*d*)

Model				u1	u2	u3	ur1	ur2	ur3
				transl. spring (kN)					
n	b	d	t	**k11**	k12	k13	k14	k15	k16
6	2	500	8	**105.791**	44.482	5.117	2.803	1.737	0.152
9	2	500	8	**78.296**	30.749	4.439	2.561	1.605	0.145
12	2	500	8	**72.540**	28.038	4.308	2.451	1.551	0.147
6	2	700	8	**68.489**	27.898	3.775	2.723	1.504	0.077
9	2	700	8	**50.357**	19.603	2.833	2.527	1.418	0.047
12	2	700	8	**46.172**	17.758	2.612	2.438	1.384	0.036
				rot. spring (kNm)					
				k41	k42	k43	**k44**	k45	k46
6	2	500	8	0.256	0.028	0.031	**82.921**	19.905	2.032
9	2	500	8	0.216	0.008	0.022	**83.259**	18.795	2.016
12	2	500	8	0.201	0.001	0.019	**82.278**	17.628	1.934
6	2	700	8	0.153	0.002	0.031	**88.187**	23.459	0.107
9	2	700	8	0.132	0.007	0.021	**87.795**	22.324	0.216
12	2	700	8	0.121	0.011	0.018	**86.456**	21.075	0.309

b	d	t	n	diff	
				k11	k44
2	500	8	6		
2	500	8	9	-25.99%	0.41%
2	500	8	12	-7.35%	-1.18%
2	700	8	6		
2	700	8	9	-26.47%	-0.44%
2	700	8	12	-8.31%	-1.53%
avg				-17.03%	-0.68%



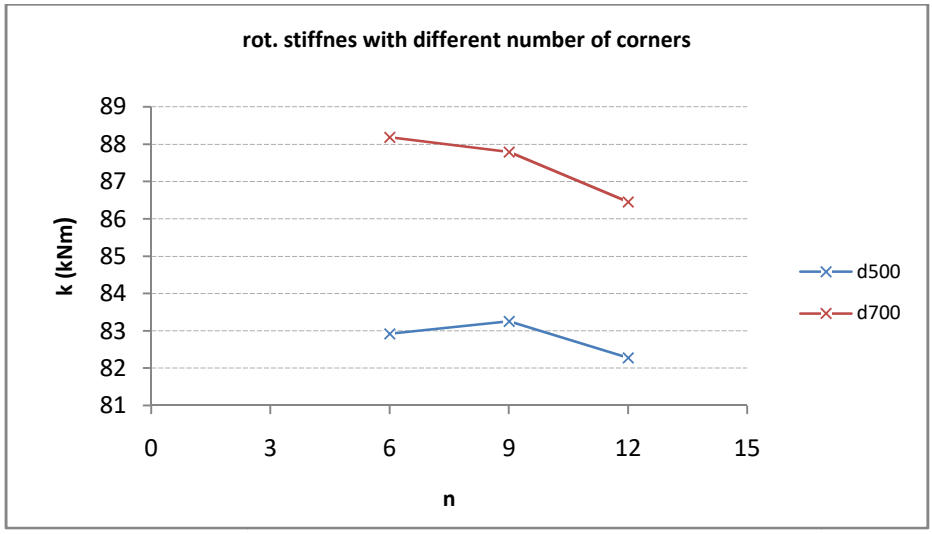


Figure 126. Stiffness with different number of corners

Von misses stresses and deformed shape at the region of load application can be seen in Figure 126 below. It is shown that as expected, the response is still in elastic range.

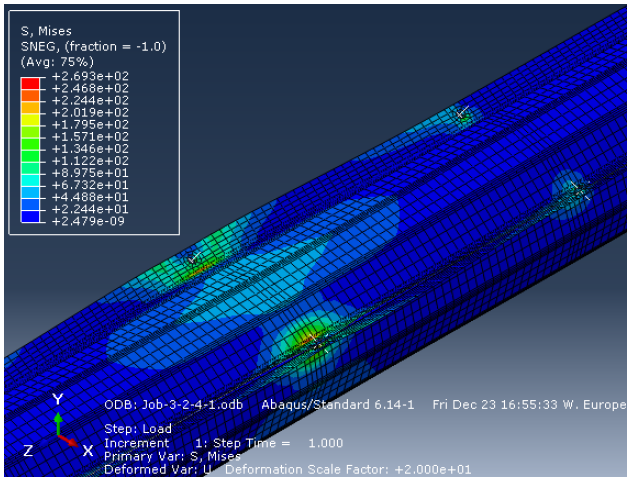


Figure 127. Von misses stress at location of load application

From the analysis above it can be conluded that parameter of diameter, thickness, and bolt spacing give significant influence on the spring stiffness of the lip connection. This method of determining connection stiffness is convenient and applicable for the need of spring values in finite strip method CUFSM, as an alternative besides experimental test.

6 CONCLUSIONS AND FURTHER WORKS

Behavior and design of thin-walled steel columns requires consideration of failure modes from local, distortional and Euler (i.e. flexural or flexural-torsional) buckling. A parametric study on the ultimate strengths of semi-closed thin-walled polygonal columns was presented in order to have those understandings on this specific type of cross-section. Finite element models were first developed in ABAQUS by using Python automation for the predefined parameters, i.e. diameter, profile slenderness, member slenderness, and bolt spacing. Issues such as application of initial geometric imperfections, material modelling, and loading conditions were addressed. Total 216 FE models were created and analysed in this parametric study. The results from FE analysis were then compared against analytical solutions carried out according to the design standard EC1993-1-3 and EC1993-1-6.

Elastic buckling analysis of FE models shows that the studied models have predominant distortional buckling as the first buckling mode. Meanwhile, some models buckled in flexural and flexural-distortional buckling mode. For this type semi-closed built-up column composed of folded plates, there is no expression in the Eurocode for predicting the elastic critical buckling, either for sectorial or global buckling modes. FE analyses need to be performed to obtain accurate elastic buckling critical load and mode shape.

From non-linear post-buckling analysis considering material, geometrical non-linearity and initial imperfections, it can be concluded that FE models in the range of the predefined parameters failed in predominant distortional mode. Models with high global slenderness, $\bar{\lambda}$=1.25, experienced distortional-flexural interaction with significantly lower resistance than the one failed in pure distortional mode. From this analysis it is suggested that for the models in this parametric study, distortional failures have lower post-buckling capacity than other failure modes, and distortional buckling may control the failure mechanism even when the elastic distortional buckling stress (f_{crd}) is higher than the elastic flexural buckling stress (f_{crf}). Moreover, it was noticed that members with high distortional and global slenderness have higher parameter sensitivity on the ultimate strength and failure mode interaction.

It is concluded that for this type of semi-closed hexagonal cross-section, bolt spacing-to-diameter ratio (s/d) of 3, 4, and 5 gives a dominated distortional post-buckling failure mode. This phenomenon occurs for member slenderness ($\bar{\lambda}$) range of 0.65 to 1.25.

Evaluation of ultimate resistance according to EN1993-1-3 shows that for members with $\bar{\lambda}$=0.65 and $\bar{\lambda}$=1.0 a good agreement was obtained, while for very slender columns $\bar{\lambda}$=1.25 a large scatter numerical results were in found in unsafe region. This corresponds to the FE results which exhibit a distortional-flexural interaction mode in those models. Interaction failure mode causes the resistance to drop. Therefore, it can be suggested that the expression used in EC1993-1-3 for reduction factor due to distortional buckling (Eq. 5.12) may be adopted for the semi-closed polygonal type of cross-section undergoes pure distortional buckling mode used in this parametric study, with provision of global slenderness, $\bar{\lambda}$< 1.25. The EC1993-1-3 became the lower bound for the results of numerical ultimate strength. It is important to note that for this type of cross-section, a careful application of design standard shall be done since current design methods ignore buckling interaction and do not

explicitly consider sectional buckling. Knowing the exact failure mode is necessary, in order to avoid too conservative predictions.

Design according to EN1993-1-6 has disagreement with the FE results with the analytical ones, especially for models with high slenderness. This result gives an insight that EN1993-1-6 overpredicted the ultimate strength for majority of the models. It is suggested that analytical resistance calculation according to EN1993-1-6 is not applicable for this type of cross section under the predefined parameters. By using this calculation for comparing the studied semi-closed polygonal profiles versus 'equivalent' cylindrical profiles, it was revealed that polygonal profiles with bolt spacing ratio, b=3 and global slenderness $\bar{\lambda}$=0.65 have higher ultimate resistances than the cylindrical ones. The higher the bolt spacing and member slenderness, the polygonal profiles tend to have declining resistance and hence pose lower capacity than the circular profiles.

Analyses of members subject to combined axial compression force and bending moment were carried out to see the effect of bending moment on ultimate resistance and influence of parameters on the N-M interaction. Three magnitude of moment, i.e. $0.05M_u$, $0.1M_u$, and $0.15M_u$ was applied. Axial-bending moment interaction curve shows that the existence of bending moment tends to decrease the ultimate strength of the member and may change the failure mode shape, however the effect was not significant.

Full factorial design with four parameters was carried out in this study in order to determine the main and interaction effects in the models. From the interaction plot it can be concluded that diameter (*d*-parameter) have the most significant influence on the ultimate strength, and followed by member slenderness (λ-parameter). Profile slenderness (*slend*-parameter) and bolt spacing (*b*-parameter) have relatively moderate influence. Interaction diameter-member slenderness and interaction member slenderness-bolt spacing are the most significant among all interactions. Profile slenderness dependency is negligible since interaction between profile slenderness with other factors were very small. Visible effect of interaction are especially shown when the factors interact with member slenderness $\bar{\lambda}$=1.25.

In terms of strength-to-weight ratio, member slenderness and diameter are most the significant parameters on it. The choice of proper geometrical properties in the design of semi-closed polygonal cross-section is of important aspect and will determine the resistance and effectiveness of the structure.

Calculations of the stiffness of lips' bolted connection were done. The results show that this spring stiffness value was mostly influenced by diameter and thickness of the profiles. The method of calculation can be applied for the purpose of required spring values for finite strip package, e.g. CUFSM. The spring values are to be fed into the software with not a straightforward way though.

To extend the work in this thesis, the following recommendations can be considered regarding further research of the semi-closed thin-walled polygonal profiles for truss columns:

1. A more extensive parametric study with larger range of parameters should be done, especially for the significant parameters. Boundary conditions, parameter of bolt spacing and member length shall be arranged to set the member for a specific failure mode, and therefore cover all possible failure modes and interaction between them.
2. Study of the effect of lips' connection stiffness at the joint area on the behaviour and design of the member.
3. Validation of the numerical analysis with experimental tests.

REFERENCES

[1] The European Wind Energy Association, "Wind in power: 2015 European Statistics," EWEA, 2016.

[2] E. Hau, *Wind Turbines: Fundamentals, Technologies, Application, Economics*. Berlin Heidelberg: Springer-Verlag, 2006.

[3] RUUKKI. (2013) RUUKKI. [Online].http://www.ruukki.com/Corporateresponsibility/Corporate-responsibility-reporting/Corporate-responsibility-report-2010/Environment/Energy-efficiency/Energy-effiency-products/Profitable-wind-power-from-on-high

[4] C. Heistermann, W. Husson, M. Limam, M. Feldmann, J. Naumes, T. Pak, T. Faber, M. Klose, K.-U. Fruhner, L. Krutschina, C. Baniotopoulos, L. Lavisas, A. Pontes, E. Ribeiro, M. Hadden, R. Sousa, L. da Silva, C. Rebelo, R. Simoes, J. Henriques M. Veljkovic, "High-strength tower in steel for wind Turbines (HISTWIN)," European Commission, Directorate-General for Research and Innovation, Brussel, 2012.

[5] M. Limam, O. Garzon, A. Olasson, F. Möller, C. Richter, D. Pak, S. Gerasimidis, E. Efthymiou, C. C. Baniotopoulos, O.-P. Holamo, R. S. Lima, C. Rebelo, P. Paulo, R. Matos, M. Silva and L. S. da Silva M. Veljkovic, "HISTWIN2 "High steel tubular towers for wind turbines" Mid-term Report," 2012.

[6] European Commission. (2016) AEOLUS4FUTURE. [Online].http://www.aeolus4future.eu/about

[7] C. Hesitermann, *Behaviour of pretensioned bolts in friction connections. Towards the use of higher strength steel in wind towers*. Luleå: Luleå University of Technology, 2011.

[8] O. Gerson, *Resistance of Polygonal Cross-Sections: Application on Steel Towers for Wind Turbines*. Luleå: Luleå University of Technology, 2013.

[9] CEN, "Eurocode 3: design of steel structures, part 1-3: general rules," European comitte for standardization, Brussels, 2002.

[10] CEN, "Eurocode 3- Design of steel structures - Part 1-5: Plated structural members," European committee for standardization, Brussels, 2009.

[11] Tomas Lyrner, Manouchehr Hassanzadeh, Thomas Stalin, John Johansson Staffan Engström, "Tall towers for large wind turbines: Report Vindforsk project V-342 Höga torn för vindkraftverk," 2010.

[12] T. Stalin and J. Johansson M. Hassanzadeh, "High towers for wind power onshore – information dissemination," Vattenfall Reasearch and Development AB, 2008.

[13] K. Hüsemann, *Ruukki Wind Towers: High truss towers for wind turbine generators*. Helsinki: RUUKKI, 2010.

[14] W. W. Yu, *Cold–Formed Steel Design*, 3rd ed. Canada: John Wiley & Sons, 2000.

[15] D. Camotim, and N. Silvestre M. Abambres, "Physically non-linear GBT analysis of thin-walled members," *Comput. Struct.*, vol. 129, pp. 148-165, 2013.

[16] J. G. and Michael, R. Teng, *Buckling of thin metal shells*. London: Spon Press, 2004.

[17] Pekoz T. Schafer B. W. (2011) Local and distortional Buckling of Cold–Formed Steel Members with Edge Stiffened Flanges. [Online]. http://ceeserver.cee.cornell.edu/tp26/TWResearchGroup/tpadd/edgepap-corr.doc

[18] Y. B. and Seo, G. H. Kwon,.: Thin-Walled Structures, 2012, pp. 126-139.

[19] G. J. Hancock, "Design for distortional buckling of flexural members," *Thin-Walled Structures, 20:3–12, Elsevier*, 1997.

[20] D. and Hancock, G. J. Yang, "Developments in design for distortional buckling of thin walled members," *ICTWS, University of Loughborough*, 2004.

[21] E. S., Batista, E. M. and Camotim, D. Dos Santos, "Experimental investigation concerning lipped channel columns undergoing local–distortional–global buckling mode interaction," *Thin-Walled Structures, 54*, pp. 19-34, 2012.

[22] E. and Tomblin, J. Barbero, "Euler Buckling of Thin-Walled Composite Columns," *Thin*

[23] N. D. and Mahendran, M. Kankanamge, "Behaviour and design of cold–formed steel beams subject to lateral–torsional buckling," *Thin-Walled Structures, 51, Elsevier Science*, pp. 25-38, 2012.

[24] Eurocode resources. (2011) [Online]. http://www.eurocode-resources.com/buckling-a16.html

[25] AISI, "Distortional buckling of cold–formed steel column," Research Report rp. 00-1 2006.

[26] Viorel Ungureanu Dan Dubina, "Erosion of Interactive Buckling Load of Thin-Walled Steel Bar Members: Contribution of "Timisoara School" ," *Ro. J. Techn. Sci. − Appl. Mechanics*, vol. 59, pp. 9–137, 2014.

[27] CEN, "Eurocode 3: Design of steel structures, part 1-8: Design of joints," European comitee fro standardization, Brussels, 2005.

[28] M. Muvafık M.E. Kartal, "Effects of Semi-Rigid Connection on Structural Responses," *Electronic Journal of Structural Engineering*, no. 10, 2010.

[29] Dassault Simulia, *Abaqus CAE Documentation: User Guide*., 2007.

[30] C.D. Moen, *Direct Strength Design for Cold-Formed Steel Members with Perforations. Ph.D. Thesis*. Baltimore: Johns Hopkins University, 2008.

ANNEXES

List of Annexes:

ANNEX A.1 MATLAB SCRIPT FOR [X, Y] DATABASE OF POLYGONALS' PROFILES

>> pcoords.m

```
  1 function [x_out, y_out, t, tg, l_lip] = pcoords(n, d, slend, fy, rcoef, nbend,↙
l_ratio, t_ratio)
  2 % Return x, y coords of points of a 1/3 of folded polygonal cross section.
  3 % input args: number of corners, CS diameter, slenderness, yield strength,
  4 % bending arc radius r/t, no. of points along the bending arcs,lip length
  5 % to diameter ratio, gusset plate thickness to sector thickness ratio.
  6 % output: [x, y], sector thickness, gussetplate thickness
  7
  8 %% Input (recomended values)
  9 % % Number of corners (entire polygon, only 3*m)
 10 % n = 9;
 11 %
 12 % % Polygon diameter
 13 % d = 500;
 14 %
 15 % % Yield strength
 16 % fy = 355;
 17 %
 18 % % Bending radius to thickness ratio
 19 % % (r/t = rcoef)
 20 % rcoef = 6;
 21 %
 22 % % Number of points along the bend
 23 % nbend = 6;
 24 %
 25 % % extension length to diameter ratio
 26 % l_ratio = 0.1;
 27 %
 28 % % Thickness of the gusset plate to sector thickness ratio
 29 % t_ratio = 1.20;
 30 %
 31 % % Slenderness
 32 % slend = 90;
 33
 34
 35 % Calculated characteristics
 36 R = d/2;
 37 epsilon = sqrt(fy/235);
 38 t = (epsilon^2 * d / slend);
 39 tg = (t_ratio*t);
 40 l_lip = l_ratio*d;
 41
 42 %% Polygon sector
 43 % Angle corresponding to one edge of the polygon
 44 theta = 2*pi/n;
 45
 46 % Angles of radii (measured from x-axis)
 47 phi=5*pi/6:-theta:pi/6;
 48
 49 % xy coords of the polygon's corners
 50 x = R*cos(phi);
 51 y = R*sin(phi);
 52
```

```
 53 %% Bends
 54 % Bending radius
 55 rbend = rcoef*t;
 56
 57 % Distance between bending centre and corner
 58 lc = rbend/cos(theta/2);
 59
 60 % Centers of bending arcs
 61 xc  = (x(2:end-1) - lc*cos(phi(2:end-1)));
 62 yc  = (y(2:end-1) - lc*sin(phi(2:end-1)));
 63
 64 % Bending arc angle
 65 theta_b = pi - theta;
 66
 67 % Angles of the edges' midlines (measured from x-axis)
 68 phi_mids = phi(1:end-1) - theta/2 ;
 69
 70 % xy coords of the arc's points
 71 for i = 1:n/3 -1;
 72     for j = 1:nbend+1;
 73         xarc(i, j) = xc(i) + rbend*cos(phi_mids(i)-(j-1)*(theta/nbend));
 74         yarc(i, j) = yc(i) + rbend*sin(phi_mids(i)-(j-1)*(theta/nbend));
 75     end;
 76 end;
 77
 78 %% Start-end extensions
 79 % Bending radius
 80 rs = rbend/5;
 81
 82 % First bend
 83 v1 = phi_mids(1)-pi/2;
 84 v2 = (phi(1)+phi_mids(1)-pi/2)/2;
 85 l1 = (t+tg)/(2*cos(phi(1)-phi_mids(1)));
 86 l2 = rs/sin(v2-phi_mids(1)+pi/2);
 87 x1 = x(1)+l1*cos(v1);
 88 y1 = y(1)+l1*sin(v1);
 89
 90 % First bend centre coords
 91 xcs(1) = x1+l2*cos(v2);
 92 ycs(1) = y1+l2*sin(v2);
 93
 94 % Last bend
 95 v1 = phi_mids(end)+pi/2;
 96 v2 = (v1+phi(end))/2;
 97 l1 = (t+tg)/(2*cos(v1-phi(end)-pi/2));
 98 l2 = rs/sin(v2-phi(end));
 99 x1 = x(end)+l1*cos(v1);
100 y1 = y(end)+l1*sin(v1);
101
102 % Last bend centre coords
103 xcs(2) = x1+l2*cos(v2);
104 ycs(2) = y1+l2*sin(v2);
105
```

```
% First and last bend arc points coords
for j = 1:nbend+1;
    xsarc(1, j) = xcs(1) + rs*cos(4*pi/3+(j-1)*((phi_mids(1)-pi/3)/nbend));
    ysarc(1, j) = ycs(1) + rs*sin(4*pi/3+(j-1)*((phi_mids(1)-pi/3)/nbend));
    xsarc(2, j) = xcs(2) + rs*cos(phi_mids(end)+pi+(j-1)*((phi(end)+pi/2-phi_mids↙
(end))/nbend));
    ysarc(2, j) = ycs(2) + rs*sin(phi_mids(end)+pi+(j-1)*((phi(end)+pi/2-phi_mids↙
(end))/nbend));
end;

%% Points of the lips
% First lip
xstart = [xsarc(1, 1) + l_lip*cos(phi(1)), xsarc(1, 1) + l_lip*cos(phi(1))/2];
ystart = [ysarc(1, 1) + l_lip*sin(phi(1)), ysarc(1, 1) + l_lip*sin(phi(1))/2];


% Last point
xend = [xsarc(2, end) + l_lip*cos(phi(end))/2, xsarc(2, end) + l_lip*cos(phi↙
(end))];
yend = [ysarc(2, end) + l_lip*sin(phi(end))/2, ysarc(2, end) + l_lip*sin(phi↙
(end))];

%% Collect the x, y values in a sorted 2xn array
xarc = xarc';
yarc = yarc';

x_out = [xstart, xsarc(1, :), xarc(:)', xsarc(2, :), xend];
y_out = [ystart, ysarc(1, :), yarc(:)', ysarc(2, :), yend];

% Plot result
% plot(x_out, y_out);

```

ANNEX A.2 MATLAB SCRIPT FOR PROFILES DATABASE AND META DATABASE

>> polygoner.m

```
  1 function [profiles, meta] = polygoner(nrange, drange, slendrange, fy, rcoef,↙
nbend, l_ratio, t_ratio, lambda);
  2 % Return a cell array with the points of all the profiles within a range of
  3 % values.
  4 % input args: numbers of corners, CS diameters, slenderness', yield strength,
  5 % bending arc radius r/t, no. of points along the bending arcs, end
  6 % extensions length, gusset plate thickness.
  7 % profiles output  : [x; y], [diameter; plate thicness; gusset plate thickness;↙
fy]
  8 % meta outpur      : d; t; tg; fy; A; Ixx; Izz; Ixz
  9
 10 % Example input
 11 % nrange = [6, 9, 12];
 12 % drange = [300:200:900];
 13 % slendrange = linspace(80, 180, 10);
 14 % lambda = [0.65, 1, 1.25];
 15 %
 16 % fy = 355;
 17 % rcoef = 6;
 18 % nbend = 4;
 19 % l_ratio = 0.1;
 20 % t_ratio = 1.2;
 21
 22 E = 210000;
 23
 24 % Initialise a cell array to host the profiles' xy values
 25 profiles = cell(length(nrange), length(drange), length(slendrange));
 26
 27 % Initialise a cell array to host the profile metadata
 28 meta = cell(length(nrange), length(drange), length(slendrange), length(lambda));
 29
 30 % Loop through the values within the given ranges
 31 for i = 1:length(nrange);
 32     for j = 1:length(drange);
 33         for k = 1:length(slendrange);
 34
 35             % Call pcoords to get data for a profile
 36             [x, y, t, tg] = pcoords(nrange(i), drange(j), slendrange(k), fy,↙
rcoef, nbend, l_ratio, t_ratio);
 37
 38             % Collect the xy values in a database
 39             profiles{i, j, k} = [x; y];
 40
 41             % Metadata of the profiles
 42             % Crate node and elem arrays for the profile appropriate for
 43             % input to cutwp_prop2 function which returns cs properties
 44
 45             % Current profile xy
 46             c_prof1 = profiles{i, j, k}';
 47
 48             % Number of vertices on the current profile
 49             l_prof = length(c_prof1);
 50
```

```
 51              % Construct the 2 extra parts by rotating the imported one
 52              R2 = [cos(-2*pi/3), -sin(-2*pi/3); sin(-2*pi/3), cos(-2*pi/3)];
 53              R3 = [cos(2*pi/3), -sin(2*pi/3); sin(2*pi/3), cos(2*pi/3)];
 54              for a = 1:l_prof;
 55                  c_prof2(a, :) = (R2*c_prof1(a, :)')';
 56                  c_prof3(a, :) = (R3*c_prof1(a, :)')';
 57              end;
 58
 59              % A column of ones
 60              col1 = ones(l_prof', 1);
 61
 62              % Construct the 'node' array
 63              node = [c_prof1(:, 1), c_prof1(:, 2);
 64                  c_prof2(:, 1), c_prof2(:, 2);
 65                  c_prof3(:, 1), c_prof3(:, 2)];
 66
 67              % Construct the 'elem' array
 68              elem = [(1:l_prof-1)', (2:l_prof)', t*ones(l_prof-1', 1);
 69                  l_prof, l_prof+1, 0.1;
 70                  l_prof+(1:l_prof-1)', l_prof+(2:l_prof)', t*ones(l_prof-1', 1);
 71                  2*l_prof, 2*l_prof+1, 0.1;
 72                  2*l_prof+(1:l_prof-1)', 2*l_prof+(2:l_prof)', t*ones(l_prof-1',↙
1);
 73                  3*l_prof, 1, 0.1];
 74
 75              % Return cs properties using cutwp
 76              [A, ~, ~, Iyy, Izz] = cutwp_prop2(node, elem);
 77
 78              % Current profile area and moment of inertia
 79              I = min(Iyy, Izz);
 80
 81              % find the element properties on the current profile
 82              nele = size(elem,1);
 83              for v = 1:nele;
 84                  sn = elem(v,1); fn = elem(v,2);
 85                  % thickness of the element
 86                  tk(v,1) = t;
 87                  % compute the coordinate of the mid point of the element
 88                  xm(v) = mean(node([sn fn],1));
 89                  ym(v) = mean(node([sn fn],2));
 90                  % compute the dimension of the element
 91                  xd(v) = diff(node([sn fn],1));
 92                  yd(v) = diff(node([sn fn],2));
 93                  % compute the length of the element
 94                  L(v,1) = norm([xd(v) yd(v)]);
 95                  Ao(v,1) = L(v)*tk(v);
 96              end
 97
 98              % Calculating cross-sectional class and effective area if needed:
 99              epsilon=sqrt(235/fy); Ep2=zeros(nele,2); lambdap=zeros(nele,1);↙
ro=zeros(nele,1);
100              for v = 1:nele;
101                  Ep = [Ao L tk];
```

```
102
103                 if Ep(v,1) == eps
104                     Ep2(v,:)=[0 123];
105                 else
106                     %EC3-1-5 Part 4.4
107                     lambdap(v)=(Ep(v,2)/Ep(v,3))/(28.4*epsilon*2);
108                     ro(v)=(lambdap(v)-0.055*4)/lambdap(v)^2;
109                     if ro(v)>1
110                         ro(v)=1;
111                     end
112                     %EC3-1-1 Table 5.2
113                     if Ep(v,2)/Ep(v,3) <= 42*epsilon
114                         Ep2(v,1)=Ep(v,1);
115                         Ep2(v,2)=3;
116                     else
117                         Ep2(v,1)=Ep(v,1)*ro(v);
118                         Ep2(v,2)=4;
119                     end
120                 end
121             end
122             % compute the effective cross section area
123             Aeff = sum(Ep2(:,1))-3*(tg+t)*t;
124             Class = max(Ep2(:,2));
125
126
127             % Classification according to EC3 1-1
128             max_side = max(sqrt(diff(node(:, 2)).^2+diff(node(:, 1)).^2))
129             if max_side/t <= 42*epsilon
130                 Class = 3;
131             else
132                 Class = 4;
133             end
134
135             % Loop through the different member slendernesses. The 'meta'
136             % array has one more dimension (4D)
137             for l = 1:length(lambda);
138
139                 % Current profile length
140                 len = lambda*pi*sqrt(E*I/(A*fy));
141
142                 % Store the metadata in a cell array
143                 meta{i, j, k, l} = [drange(j); t; tg; fy; A; Iyy; Izz; len(l);↙
Aeff; Class];
144             end
145         end
146     end
147 end
148
149
150 % Plot result
151 % plot(x, y);
152
153 % Save the profile database and metadata to the current directory as .mat
```

```
154 save('profiles.mat', 'profiles');
155 save('meta.mat', 'meta');
156
```

ANNEX A.3 MATLAB SCRIPT FOR FACTORIAL DESIGN IN FE PARAMETRIC STUDIES

```
% Full factorial design, 4 factors

parameters = {
%    'Number of corners'
    'Diameter'
    'Profile slenderness'
    'Member slenderness'
    'Bolt spacing'
    }';
% Create generators, design space and confounding
dff = fullfact([3 2 3 3]);

% Create the generators with the real values of the factors
dff2 = zeros(54, 4);
dff2(dff(:, 1)==1, 1) = 500;
dff2(dff(:, 1)==2, 1) = 700;
dff2(dff(:, 1)==3, 1) = 900;
dff2(dff(:, 2)==1, 2) = 58;
dff2(dff(:, 2)==2, 2) = 70;
dff2(dff(:, 3)==1, 3) = 0.65;
dff2(dff(:, 3)==2, 3) = 1;
dff2(dff(:, 3)==3, 3) = 1.25;
dff2(dff(:, 4)==1, 4) = 3;
dff2(dff(:, 4)==2, 4) = 4;
dff2(dff(:, 4)==3, 4) = 5;


% Import max load results
filename = 'C:\Users\bona\parametric\maxforcedispldata-N.txt';
delimiter = '\t';
formatSpec = '%*s%*s%*s%*s%f%*s%*s%[^\n\r]';
fileID = fopen(filename,'r');
dataArray = textscan(fileID, formatSpec, 'Delimiter', delimiter,  'ReturnOnError',↙
false);
fclose(fileID);
maxload = [dataArray{1:end-1}];
clearvars filename delimiter formatSpec fileID dataArray;

% Plot the interactions
figure;
interactionplot(maxload', dff2, 'varnames', parameters);

```

```
import numpy as np
import string
import sys
import os
from part import *
from material import *
from section import *
from assembly import *
from step import *
from interaction import *
from load import *
from mesh import *
from optimization import *
from job import *
from sketch import *
#from visualization import *
from connectorBehavior import *
session.journalOptions.setValues(replayGeometry=COORDINATE,
recoverGeometry=COORDINATE)
# Import profiles database and profile metadata
------------------------------------------------------------

# Import pickle to load the .pkl database
import pickle

# Define a method to get the block number of a specific string in the
keywords

def GetBlockPosition(model,blockPrefix):
    pos = 0
    for block in model.keywordBlock.sieBlocks:
        if
        string.lower(block[0:len(blockPrefix)])==string.lower(blockPrefix):
            return pos
        pos=pos+1
    return -1

# Open and read the database
profiles_file = open("./profiles.pkl",'rb')
profiles = pickle.load(profiles_file)
profiles_file.close()

profiles_file = open("./meta.pkl",'rb')
profiles_meta = pickle.load(profiles_file)
profiles_file.close()

# number of corners
#i = int(sys.argv[-5])
i = 0

# diameter of the profile
#j = int(sys.argv[-4])
j = 0

# Profile slenderness
#k = int(sys.argv[-3])
k = 0

# Member slenderness
#l = int(sys.argv[-2])
l = 0

```

```
59  # bolt spacing to diameter ratio (s/d)
60  #b = int(sys.argv[-1])
61  b = 6
62
63  # Variables holding information of the current profile
64  buckle_model = 'BCKL-'+str(i+1)+'-'+str(j+1)+'-'+str(k+1)+'-'+str(l+1)
65  current_d = float(profiles_meta[i][j][k][l][0][0])
66  current_t = float(profiles_meta[i][j][k][l][1][0])
67  current_tg = float(profiles_meta[i][j][k][l][2][0])
68  current_fy = float(profiles_meta[i][j][k][l][3][0])
69  current_l = float(profiles_meta[i][j][k][l][7][0])
70  current_llip =
    sqrt((profiles[i][j][k][0][0]-profiles[i][j][k][0][2])**2+(profiles[i][j][k]
    [1][0]-profiles[i][j][k][1][2])**2)
71  area = profiles_meta[i][j][k][l][4][0]
72  current_Iy = float(profiles_meta[i][j][k][l][5][0])
73
74  # Buckling model
    ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
    ++++++++++++++++++++++++++++++++++++
75
76  mdb.Model(modelType=STANDARD_EXPLICIT, name=buckle_model)
77  c_model = mdb.models[buckle_model]
78
79
80  # Delete initial model
81  del mdb.models['Model-1']
82
83  # Create Parts
    ----------------------------------------------------------------------------
    ----------------------------------------
84
85  # Sector
86
87  # -Profile sketch for sector
88  sector_sketch = c_model.ConstrainedSketch(name='sector', sheetSize=1200.0)
89
90  # -Sketch sector lines
91  for n in range(profiles[i][j][k].shape[1]-1):
92      sector_sketch.Line(
93          point1=(profiles[i][j][k][0][n], profiles[i][j][k][1][n]),
94          point2=(profiles[i][j][k][0][n+1], profiles[i][j][k][1][n+1])
95          )
96
97  # -Extrude sector part
98  l_tot = 2*current_l + 3*current_d
99  sector_part = c_model.Part(
100     dimensionality=THREE_D,
101     name='sector',
102     type=DEFORMABLE_BODY
103     )
104 sector_part.BaseShellExtrude(
105     depth=l_tot,
106     sketch=sector_sketch
107     )
108
109 # Calculate bolt positions
110
111 # -Distance on the width
112 bolts_w = current_llip/2
113
114 # -Distances on the length
```

```
115   current_b =  b
116   s = current_b*current_d
117   (n0, s0) = divmod(current_l, s)
118   s1 = (s0 + s)/2
119
120   bolts_z1 = np.concatenate([[bolts_w], bolts_w + ((current_d -
      current_llip)/5) * np.linspace(1, 4, 4), [current_d - bolts_w]])
121   bolts_z2 = np.concatenate([[current_d + s1], s1 + current_d + (s *
      np.linspace(1, n0-1, n0-1))])
122   bolts_z3 = bolts_z1 + (current_l + current_d)
123   bolts_z4 = bolts_z2 + (current_l + current_d)
124   bolts_z5 = bolts_z3 + (current_l + current_d)
125
126   bolts_z = np.concatenate([bolts_z1, bolts_z2, bolts_z3, bolts_z4, bolts_z5])
127
128   # Washer diameter
129   d_washer = 30
130
131   # Initiate list to store datum planes
132   datum_p=[]
133
134   # Make holes
135
136   for o in range(int(bolts_z.shape[0])):
137
138       sector_part.HoleBlindFromEdges(
139           depth=1.0,
140           diameter=d_washer,
141           distance1=bolts_z[o],
142           distance2=bolts_w,
143
              edge1=sector_part.edges.getClosest(coordinates=((profiles[i][j][k][0
              ][1], profiles[i][j][k][1][1], 0),))[0][0],
144
              edge2=sector_part.edges.getClosest(coordinates=((profiles[i][j][k][0
              ][0], profiles[i][j][k][1][0], 1),))[0][0],
145
              plane=sector_part.faces.getClosest(coordinates=((profiles[i][j][k][0
              ][0], profiles[i][j][k][1][0], 0),))[0][0],
146           planeSide=SIDE1
147           )
148
149       sector_part.HoleBlindFromEdges(
150           depth=1.0,
151           diameter=d_washer,
152           distance1=bolts_z[o],
153           distance2=bolts_w,
154
              edge1=sector_part.edges.getClosest(coordinates=((profiles[i][j][k][0
              ][-2], profiles[i][j][k][1][-2], 0),))[0][0],
155
              edge2=sector_part.edges.getClosest(coordinates=((profiles[i][j][k][0
              ][-1], profiles[i][j][k][1][-1], 1),))[0][0],
156
              plane=sector_part.faces.getClosest(coordinates=((profiles[i][j][k][0
              ][-1], profiles[i][j][k][1][-1], 0),))[0][0],
157           planeSide=SIDE1
158           )
159
160       # Create datum planes to be used for partitioning the sector
161
162       datum1=sector_part.DatumPlaneByPrincipalPlane(
```

```
163             offset=bolts_z[o]-bolts_w,
164             principalPlane=XYPLANE
165             )
166         datum2=sector_part.DatumPlaneByPrincipalPlane(
167             offset=bolts_z[o]+bolts_w,
168             principalPlane=XYPLANE
169             )
170         datum_p.append(datum1)
171         datum_p.append(datum2)
172 
173     # Partition the sector
174 
175     # -Number of datum planes
176     n_dat = int(len(sector_part.datums))
177 
178     # cut all the faces using the datum planes
179 
180     for o in range((n_dat-2)):
181     #for o in range(2):
182         sector_part.PartitionFaceByDatumPlane(
183             datumPlane=sector_part.datums.items()[o+1][1],
184             faces=sector_part.faces[:]
185             )
186 
187     # Gusset
188 
189     # -Profile sketch for gusset
190     gusset_sketch=c_model.ConstrainedSketch(name='__profile__',
        sheetSize=1200.0)
191 
192     # -Sketch gusset lines
193     # First point of the first sector
194     x0 = profiles[i][j][k][0][0]
195     y0 = profiles[i][j][k][1][0]
196 
197     # Angle of the first gusset fin
198     phi = pi*5/6
199 
200     # Calculate the end point of the gusset's first fin as an orthogonal
        projection of the sector's first point on the line of the gusset plate
201     gp1 = np.array([(x0*cos(phi)+y0*sin(phi))*cos(phi),
        (x0*cos(phi)+y0*sin(phi))*sin(phi)])
202 
203     # Rotation matrix to multiply the previous point in order to get the
        points of the other 2 gusset fins
204     Rmat = np.array([[cos(-2*pi/3), -sin(-2*pi/3)], [sin(-2*pi/3),
        cos(-2*pi/3)]])
205 
206     # Calculate the end points of the other 2 gusset fins by multiplying with
        the 120 degrees rotation matrix
207     gp2 = gp1.dot(Rmat)
208     gp3 = gp2.dot(Rmat)
209 
210     # Draw lines for the sketch of the gusset plate between 0, 0 and the
        calculated points gp1, gp2, gp3
211     gusset_sketch.Line(
212         point1=(0.0, 0.0),
213         point2=(gp1[0], gp1[1])
214         )
215     gusset_sketch.Line(
216         point1=(0.0, 0.0),
217         point2=(gp2[0], gp2[1])
```

```
218         )
219     gusset_sketch.Line(
220         point1=(0.0, 0.0),
221         point2=(gp3[0], gp3[1])
222         )
223 
224     # -Extrude gusset part
225     gusset_part=c_model.Part(
226         dimensionality=THREE_D,
227         name='gusset',
228         type=DEFORMABLE_BODY
229         )
230     gusset_part.BaseShellExtrude(
231         depth=current_d,
232         sketch=gusset_sketch
233         )
234 
235     # -Holes
236     for o in range(int(bolts_z1.shape[0])):
237         gusset_part.HoleBlindFromEdges(
238             depth=1.0,
239             diameter=d_washer,
240             distance1=bolts_z1[o],
241             distance2=bolts_w,
242             edge1=gusset_part.edges.getClosest(coordinates=((gp1[0]/2,
                gp1[1]/2, 0),))[0][0],
243             edge2=gusset_part.edges.getClosest(coordinates=((gp1[0], gp1[1],
                1),))[0][0],
244             plane=gusset_part.faces.getClosest(coordinates=((gp1[0], gp1[1],
                0),))[0][0],
245             planeSide=SIDE1
246             )
247 
248         gusset_part.HoleBlindFromEdges(
249             depth=1.0,
250             diameter=d_washer,
251             distance1=bolts_z1[o],
252             distance2=bolts_w,
253             edge1=gusset_part.edges.getClosest(coordinates=((gp2[0]/2,
                gp2[1]/2, 0),))[0][0],
254             edge2=gusset_part.edges.getClosest(coordinates=((gp2[0], gp2[1],
                1),))[0][0],
255             plane=gusset_part.faces.getClosest(coordinates=((gp2[0], gp2[1],
                0),))[0][0],
256             planeSide=SIDE1
257             )
258 
259         gusset_part.HoleBlindFromEdges(
260             depth=1.0,
261             diameter=d_washer,
262             distance1=bolts_z1[o],
263             distance2=bolts_w,
264             edge1=gusset_part.edges.getClosest(coordinates=((gp3[0]/2,
                gp3[1]/2, 0),))[0][0],
265             edge2=gusset_part.edges.getClosest(coordinates=((gp3[0], gp3[1],
                1),))[0][0],
266             plane=gusset_part.faces.getClosest(coordinates=((gp3[0], gp3[1],
                0),))[0][0],
267             planeSide=SIDE1
268             )
269 
270     # Partition gusset
```

```
271
272   gusset_part.DatumPointByCoordinate((gp1[0]-current_llip*cos(5*pi/6),
      gp1[1]-current_llip*sin(5*pi/6), 0),)
273   gusset_part.DatumPointByCoordinate((gp1[0]-current_llip*cos(5*pi/6),
      gp1[1]-current_llip*sin(5*pi/6), current_d),)
274
275   gusset_part.PartitionFaceByShortestPath(
276       faces=gusset_part.faces.getClosest(coordinates=((gp1[0], gp1[1],
          0),))[0][0],
277       point1=gusset_part.datum.items()[0][1],
278       point2=gusset_part.datum.items()[1][1],
279       )
280
281   gusset_part.DatumPointByCoordinate((gp2[0]-current_llip*cos(-pi/2),
      gp2[1]-current_llip*sin(-pi/2), 0),)
282   gusset_part.DatumPointByCoordinate((gp2[0]-current_llip*cos(-pi/2),
      gp2[1]-current_llip*sin(-pi/2), current_d),)
283
284   gusset_part.PartitionFaceByShortestPath(
285       faces=gusset_part.faces.getClosest(coordinates=((gp2[0], gp2[1],
          0),))[0][0],
286       point1=gusset_part.datum.items()[2][1],
287       point2=gusset_part.datum.items()[3][1],
288       )
289
290   gusset_part.DatumPointByCoordinate((gp3[0]-current_llip*cos(pi/6),
      gp3[1]-current_llip*sin(pi/6), 0),)
291   gusset_part.DatumPointByCoordinate((gp3[0]-current_llip*cos(pi/6),
      gp3[1]-current_llip*sin(pi/6), current_d),)
292
293   gusset_part.PartitionFaceByShortestPath(
294       faces=gusset_part.faces.getClosest(coordinates=((gp3[0], gp3[1],
          0),))[0][0],
295       point1=gusset_part.datum.items()[4][1],
296       point2=gusset_part.datum.items()[5][1],
297       )
298
299   # Material
      ------------------------------------------------------------------------
      ----------------------------------------
300
301   c_model.Material(name='optim355')
302   c_model.materials['optim355'].Elastic(table=((210000.0, 0.3), ))
303
304   # Create sections
      ------------------------------------------------------------------------
      ----------------------------------
305
306   # -for sector
307   c_model.HomogeneousShellSection(
308       idealization=NO_IDEALIZATION,
309       integrationRule=SIMPSON,
310       material='optim355',
311       name='sector',
312       numIntPts=5,
313       poissonDefinition=DEFAULT,
314       preIntegrate=OFF,
315       temperature=GRADIENT,
316       thickness=current_t,
317       thicknessField='',
318       thicknessModulus=None,
319       thicknessType=UNIFORM,
```

```
320         useDensity=OFF
321         )
322 
323     # -for gusset
324     c_model.HomogeneousShellSection(
325         idealization=NO_IDEALIZATION,
326         integrationRule=SIMPSON,
327         material='optim355',
328         name='gusset',
329         numIntPts=5,
330         poissonDefinition=DEFAULT,
331         preIntegrate=OFF,
332         temperature=GRADIENT,
333         thickness=current_tg,
334         thicknessField='',
335         thicknessModulus=None,
336         thicknessType=UNIFORM,
337         useDensity=OFF
338         )
339 
340     # Assign sections
        ------------------------------------------------------------------------
        ---------------------------------
341 
342     # -for sector
343     sector_part.Set(
344         faces=sector_part.faces[:],
345         name='AllSectorFaces'
346         )
347     sector_part.SectionAssignment(
348         offset=0.0,
349         offsetField='',
350         offsetType=MIDDLE_SURFACE,
351         region=sector_part.sets['AllSectorFaces'],
352         sectionName='sector',
353         thicknessAssignment=FROM_SECTION
354         )
355 
356     # -for gusset
357     gusset_part.Set(
358         faces=gusset_part.faces[:],
359         name='AllGussetFaces')
360     gusset_part.SectionAssignment(
361         offset=0.0,
362         offsetField='',
363         offsetType=MIDDLE_SURFACE,
364         region=gusset_part.sets['AllGussetFaces'],
365         sectionName='gusset',
366         thicknessAssignment=FROM_SECTION
367         )
368 
369     # Meshing
        ------------------------------------------------------------------------
        ----------------------------------------
370 
371     # Global seeding in mm
372     seedsize = 30
373 
374     # -Sector
375     sector_part.setMeshControls(
376         algorithm=MEDIAL_AXIS,
377         elemShape=QUAD,
```

```
378         regions=sector_part.faces[:]
379         )
380     sector_part.seedPart(
381         deviationFactor=0.1,
382         minSizeFactor=0.1,
383         size=seedsize
384         )
385     sector_part.generateMesh()
386 
387     # -Gusset
388     gusset_part.setMeshControls(
389         algorithm=MEDIAL_AXIS,
390         elemShape=QUAD,
391         regions=gusset_part.faces[:]
392         )
393     gusset_part.seedPart(
394         deviationFactor=0.1,
395         minSizeFactor=0.1,
396         size=seedsize
397         )
398     gusset_part.generateMesh()
399 
400     # Create assembly
      ------------------------------------------------------------------------
      ----------------------------------
401 
402     c_assembly=c_model.rootAssembly
403     c_assembly.DatumCsysByDefault(CARTESIAN)
404 
405     # -Sectors
406     s1_instance=c_assembly.Instance(
407         dependent=ON,
408         name='sector-1',
409         part=sector_part
410         )
411     c_assembly.DatumAxisByPrincipalAxis(
412         principalAxis=ZAXIS
413         )
414     s3_instance=c_assembly.RadialInstancePattern(
415         axis=(0.0, 0.0, 1.0),
416         instanceList=('sector-1', ),
417         number=3, point=(0.0, 0.0, 0.0),
418         totalAngle=240.0
419         )
420 
421     s2_instance=s3_instance[0]
422     s3_instance=s3_instance[1]
423 
424     s_instance = (s1_instance,s2_instance ,s3_instance)
425 
426     # -Gusset plate
427 
428     # --Create the instances
429     g1_instance=c_assembly.Instance(
430         dependent=ON,
431         name='gusset-1',
432         part=gusset_part
433         )
434     g2_instance=c_assembly.Instance(
435         dependent=ON,
436         name='gusset-2',
437         part=gusset_part
```

```
438         )
439     g3_instance=c_assembly.Instance(
440         dependent=ON,
441         name='gusset-3',
442         part=gusset_part
443         )
444 
445     # --Translate them to the right position
446     g2_instance.translate(
447         vector=(0.0, 0.0, (current_l + current_d))
448         )
449     g3_instance.translate(
450         vector=(0.0, 0.0, 2*(current_l + current_d))
451         )
452 
453     # Interactions
        ------------------------------------------------------------------------
        -------------------------------------
454 
455     # Create sets node regions to be used for the tie and coupling constraints
456     # initiate variables to store points for findAt
457 
458     holes11=()
459     holes12=()
460     holes21=()
461     holes22=()
462     holes31=()
463     holes32=()
464     gholes1=()
465     gholes2=()
466     gholes3=()
467 
468     # Position of the holes on the cross-section
469     sh11 = np.array([profiles[i][j][k][0][1], profiles[i][j][k][1][1]])
470     sh12 = np.array([profiles[i][j][k][0][-2], profiles[i][j][k][1][-2]])
471 
472     gh1 = (gp1[0]-bolts_w*cos(5*pi/6), gp1[1]-bolts_w*sin(5*pi/6))
473     gh2 = (gp2[0]-bolts_w*cos(-pi/2), gp2[1]-bolts_w*sin(-pi/2))
474     gh3 = (gp3[0]-bolts_w*cos(pi/6), gp3[1]-bolts_w*sin(pi/6))
475 
476     gh=(gh1, gh2, gh3)
477 
478     # Rotation matrix to multiply the previous point in order to get the
        points of the other 2 gusset fins
479     Rmat = np.array([[cos(-2*pi/3), -sin(-2*pi/3)], [sin(-2*pi/3),
        cos(-2*pi/3)]])
480 
481     # Calculate the end points of the other 2 gusset fins by multiplying with
        the 120 degrees rotation matrix
482     sh21 = sh11.dot(Rmat)
483     sh22 = sh12.dot(Rmat)
484 
485     sh31 = sh21.dot(Rmat)
486     sh32 = sh22.dot(Rmat)
487 
488     sh = ((sh11, sh12), (sh21, sh22), (sh31, sh32))
489 
490     # Create reference points for the bolt ridig body couplings
491 
492     # Create the necessary sets and the tie constraints for all the bolts
493 
494     # End 1 connection
```

```
495   for oo in (range(3)):
496       ii=1
497       for o in tuple(bolts_z1):
498
499           c_assembly.ReferencePoint((gh[oo-3][0], gh[oo-3][1], float(o)))
500
501           c_assembly.Set(
502               edges=s_instance[oo-3].edges.findAt(((sh[oo-3][0][0],
                  sh[oo-3][0][1], float(o)-d_washer/2), ), )+\
503               s_instance[oo-2].edges.findAt(((sh[oo-2][1][0],
                  sh[oo-2][1][1], float(o)-d_washer/2), ), )+\
504               g1_instance.edges.findAt(((gh[oo-3][0], gh[oo-3][1],
                  float(o)-d_washer/2), ), ),
505               name='b'+str(ii)+str(oo)+'set1'
506               )
507
508           c_model.RigidBody(
509               name='b1'+str(ii)+str(oo)+'joint1',
510
                  refPointRegion=Region(referencePoints=(c_assembly.referencePoint
                  s.findAt((gh[oo-3][0], gh[oo-3][1], float(o))), )),
511               tieRegion=c_assembly.sets['b'+str(ii)+str(oo)+'set1']
512               )
513
514           ii+=1
515
516   # Span 1
517
518   for oo in (range(3)):
519       ii=1
520       for o in tuple(bolts_z2):
521
522           c_assembly.ReferencePoint((gh[oo-3][0], gh[oo-3][1], float(o)))
523
524           c_assembly.Set(
525               edges=s_instance[oo-3].edges.findAt(((sh[oo-3][0][0],
                  sh[oo-3][0][1], float(o)-d_washer/2), ), )+\
526               s_instance[oo-2].edges.findAt(((sh[oo-2][1][0],
                  sh[oo-2][1][1], float(o)-d_washer/2), ), ),
527               name='b'+str(ii)+str(oo)+'-set2'
528               )
529
530           c_model.RigidBody(
531               name='b1'+str(ii)+str(oo)+'span1',
532
                  refPointRegion=Region(referencePoints=(c_assembly.referencePoint
                  s.findAt((gh[oo-3][0], gh[oo-3][1], float(o))), )),
533               tieRegion=c_assembly.sets['b'+str(ii)+str(oo)+'-set2']
534               )
535
536           ii+=1
537
538   # middle connection
539
540
541   for oo in (range(3)):
542       ii=1
543       for o in tuple(bolts_z3):
544
545           c_assembly.ReferencePoint((gh[oo-3][0], gh[oo-3][1], float(o)))
546
547           c_assembly.Set(
```

```
548             edges=s_instance[oo-3].edges.findAt(((sh[oo-3][0][0],
                sh[oo-3][0][1], float(o)-d_washer/2), ), )+\
549             s_instance[oo-2].edges.findAt(((sh[oo-2][1][0],
                sh[oo-2][1][1], float(o)-d_washer/2), ), )+\
550             g2_instance.edges.findAt(((gh[oo-3][0], gh[oo-3][1],
                float(o)-d_washer/2), ), ),
551             name='b'+str(ii)+str(oo)+'set3'
552             )
553
554         c_model.RigidBody(
555             name='b1'+str(ii)+str(oo)+'joint2',
556
                refPointRegion=Region(referencePoints=(c_assembly.referencePoint
                s.findAt((gh[oo-3][0], gh[oo-3][1], float(o))), )),
557             tieRegion=c_assembly.sets['b'+str(ii)+str(oo)+'set3']
558             )
559
560         ii+=1
561
562 # Span 2
563
564 for oo in (range(3)):
565     ii=1
566     for o in tuple(bolts_z4):
567
568         c_assembly.ReferencePoint((gh[oo-3][0], gh[oo-3][1], float(o)))
569
570         c_assembly.Set(
571             edges=s_instance[oo-3].edges.findAt(((sh[oo-3][0][0],
                sh[oo-3][0][1], float(o)-d_washer/2), ), )+\
572             s_instance[oo-2].edges.findAt(((sh[oo-2][1][0],
                sh[oo-2][1][1], float(o)-d_washer/2), ), ),
573             name='b'+str(ii)+str(oo)+'-set4'
574             )
575
576         c_model.RigidBody(
577             name='b1'+str(ii)+str(oo)+'span2',
578
                refPointRegion=Region(referencePoints=(c_assembly.referencePoint
                s.findAt((gh[oo-3][0], gh[oo-3][1], float(o))), )),
579             tieRegion=c_assembly.sets['b'+str(ii)+str(oo)+'-set4']
580             )
581
582         ii+=1
583
584 # End 2 connection
585 for oo in (range(3)):
586     ii=1
587     for o in tuple(bolts_z5):
588
589         c_assembly.ReferencePoint((gh[oo-3][0], gh[oo-3][1], float(o)))
590
591         c_assembly.Set(
592             edges=s_instance[oo-3].edges.findAt(((sh[oo-3][0][0],
                sh[oo-3][0][1], float(o)-d_washer/2), ), )+\
593             s_instance[oo-2].edges.findAt(((sh[oo-2][1][0],
                sh[oo-2][1][1], float(o)-d_washer/2), ), )+\
594             g3_instance.edges.findAt(((gh[oo-3][0], gh[oo-3][1],
                float(o)-d_washer/2), ), ),
595             name='b'+str(ii)+str(oo)+'set5'
596             )
597
```

```
598             c_model.RigidBody(
599                 name='b1'+str(ii)+str(oo)+'joint3',
600 
                    refPointRegion=Region(referencePoints=(c_assembly.referencePoint
                    s.findAt((gh[oo-3][0], gh[oo-3][1], float(o))), )),
601                 tieRegion=c_assembly.sets['b'+str(ii)+str(oo)+'set5']
602                 )
603 
604             ii+=1
605 
606     # Create reference points for BCs/loads.
607 
608     # -RPs for the faces at the two ends of the columns
609     c_assembly.ReferencePoint((0.0, 0.0, 0.0))
610     c_assembly.ReferencePoint((0.0, 0.0, (2*current_l + 3*current_d)))
611 
612     # - RP at the middle
613     c_assembly.ReferencePoint((0.0, 0.0, (current_l + 1.5*current_d)))
614 
615 
616     # - End face couplings to reference points
617 
618     # End 1
619     c_assembly.Set(
620         name='RP-1-set',
621         referencePoints=(c_assembly.referencePoints.findAt((0, 0, 0)), )
622         )
623 
624     c_assembly.Set(
625 
            edges=g1_instance.edges.getByBoundingBox(-current_d,-current_d,0,current
            _d,current_d,0)+\
626 
            s_instance[0].edges.getByBoundingBox(-current_d,-current_d,0,current_d,c
            urrent_d,0)+\
627 
            s_instance[1].edges.getByBoundingBox(-current_d,-current_d,0,current_d,c
            urrent_d,0)+\
628 
            s_instance[2].edges.getByBoundingBox(-current_d,-current_d,0,current_d,c
            urrent_d,0),
629         name='end1-face',
630         )
631 
632     c_model.Coupling(
633         controlPoint=c_assembly.sets['RP-1-set'],
634         couplingType=KINEMATIC,
635         influenceRadius=WHOLE_SURFACE,
636         localCsys=None,
637         name='end1-coupling',
638         surface=c_assembly.sets['end1-face'],
639         u1=ON, u2=ON, u3=ON, ur1=ON, ur2=ON, ur3=ON
640         )
641 
642     # End 2
643 
644     c_assembly.Set(
645         name='RP-2-set',
646         referencePoints=(c_assembly.referencePoints.findAt((0, 0,
            2*(current_l+1.5*current_d))), )
647         )
648 
```

```
649   c_assembly.Set(
650       edges=g3_instance.edges.getByBoundingBox(-current_d,-current_d,2*(curren
          t_l+1.5*current_d),current_d,current_d,2*(current_l+1.5*current_d))+\
651       s_instance[0].edges.getByBoundingBox(-current_d,-current_d,2*(current_l+
          1.5*current_d),current_d,current_d,2*(current_l+1.5*current_d))+\
652       s_instance[1].edges.getByBoundingBox(-current_d,-current_d,2*(current_l+
          1.5*current_d),current_d,current_d,2*(current_l+1.5*current_d))+\
653       s_instance[2].edges.getByBoundingBox(-current_d,-current_d,2*(current_l+
          1.5*current_d),current_d,current_d,2*(current_l+1.5*current_d)),
654       name='end2-face'
655       )
656
657   c_model.Coupling(
658       controlPoint=c_assembly.sets['RP-2-set'],
659       couplingType=KINEMATIC, influenceRadius=WHOLE_SURFACE,
660       localCsys=None,
661       name='end2-coupling',
662       surface=c_assembly.sets['end2-face'],
663       u1=ON, u2=ON, u3=ON, ur1=ON, ur2=ON, ur3=ON
664       )
665
666   # Middle
667
668   c_assembly.Set(
669       name='RP-Mid-set',
670       referencePoints=(c_assembly.referencePoints.findAt((0.0, 0.0,
          (current_l + 1.5*current_d))), )
671       )
672
673   c_assembly.Set(
674       edges=g2_instance.edges.findAt(((0, 0, (current_l + 1.5*current_d)),
          ), ),
675       name='gusset-fin-interface',
676       )
677
678   c_model.Coupling(
679       controlPoint=c_assembly.sets['RP-Mid-set'],
680       couplingType=KINEMATIC,
681       influenceRadius=WHOLE_SURFACE,
682       localCsys=None,
683       name='Mid-coupling',
684       surface=c_assembly.sets['gusset-fin-interface'],
685       u1=ON, u2=ON, u3=ON, ur1=ON, ur2=ON, ur3=ON
686       )
687
688   # Step
      -------------------------------------------------------------------------
      -------
689
690   c_model.BuckleStep(
691       maxIterations=300,
692       name='Buckling',
693       numEigen=4,
694       previous='Initial',
695       vectors=10
696       )
697
698   # Boundary Conditions
```

```
    ------------------------------------------------------------
699
700 # BCs
701 end1_BC=c_model.DisplacementBC(
702     amplitude=UNSET,
703     createStepName='Initial',
704     distributionType=UNIFORM,
705     fieldName='',
706     localCsys=None,
707     name='fix-end1',
708     region=Region(referencePoints=(c_assembly.referencePoints.findAt((0,
        0, 0)), )),
709     u1=SET, u2=SET, u3=SET, ur1=UNSET, ur2=UNSET, ur3=SET
710     )
711
712 end2_BC=c_model.DisplacementBC(
713     amplitude=UNSET,
714     createStepName='Initial',
715     distributionType=UNIFORM,
716     fieldName='',
717     localCsys=None,
718     name='fix-end2',
719     region=Region(referencePoints=(c_assembly.referencePoints.findAt((0,
        0, 2*(current_l+1.5*current_d))), )),
720     u1=SET, u2=SET, u3=UNSET, ur1=UNSET, ur2=UNSET, ur3=SET
721     )
722
723 middle_BC=c_model.DisplacementBC(
724     amplitude=UNSET,
725     createStepName='Initial',
726     distributionType=UNIFORM,
727     fieldName='',
728     localCsys=None,
729     name='fix-middle',
730     region=Region(referencePoints=(c_assembly.referencePoints.findAt((0,
        0, current_l+1.5*current_d)), )),
731     u1=SET, u2=SET, u3=UNSET, ur1=UNSET, ur2=UNSET, ur3=UNSET
732     )
733
734
735 # Load
    ------------------------------------------------------------------------
    ---
736
737 c_model.ConcentratedForce(
738     cf3=-1.0,
739     createStepName='Buckling',
740     distributionType=UNIFORM,
741     field='',
742     localCsys=None,
743     name='compression',
744     region=c_assembly.sets['RP-2-set']
745     )
746
747 # Create the job
    ------------------------------------------------------------------
748
749 c_job=mdb.Job(
750     atTime=None,
751     contactPrint=OFF,
752     description='',
753     echoPrint=OFF,
```

```
754         explicitPrecision=SINGLE,
755         getMemoryFromAnalysis=True,
756         historyPrint=OFF,
757         memory=90,
758         memoryUnits=PERCENTAGE,
759         model=buckle_model,
760         modelPrint=OFF,
761         multiprocessingMode=DEFAULT,
762         name=buckle_model,
763         nodalOutputPrecision=SINGLE,
764         numCpus=1,
765         numGPUs=0,
766         queue=None,
767         resultsFormat=ODB,
768         scratch='',
769         type=ANALYSIS,
770         userSubroutine='',
771         waitHours=0,
772         waitMinutes=0
773         )
774
775     # Edit the keywords to output translations on the output file
776     c_model.keywordBlock.synchVersions(storeNodesAndElements=False)
777     c_model.keywordBlock.insert(GetBlockPosition(c_model,'*End Step')-1,
        '*NODE FILE\nU')
778
779     # Write the input file
780     mdb.jobs[buckle_model].writeInput()
781
782     # RIKS model, Only axial
        ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
        ++++++++++++++++++++++++++++
783
784     riks_model_N = 'RIKS-N-'+str(i+1)+'-'+str(j+1)+'-'+str(k+1)+'-'+str(l+1)
785
786     # Copy model from buckling analysis
787     r_model_N=mdb.Model(
788         name=riks_model_N,
789         objectToCopy=c_model
790         )
791
792     # Delete buckling step
793     del r_model_N.steps['Buckling']
794
795     # Create RIKS step
796     r_model_N.StaticRiksStep(
797         name='RIKS',
798         previous='Initial',
799         nlgeom=ON,
800         maxNumInc=30,
801         initialArcInc=0.2
802         )
803
804     # Change to plastic material, optim355
805     r_model_N.materials['optim355'].Plastic(table=((381.1, 0.0), (
806         391.2, 0.0053), (404.8, 0.0197), (418.0, 0.0228), (444.2, 0.0310),
            (499.8,
807         0.0503), (539.1, 0.0764), (562.1, 0.1009), (584.6, 0.1221), (594.4,
808         0.1394)))
809
810     # Apply concentrated force
811     N_pl_rd = 510*area
```

```
812
813   r_model_N.ConcentratedForce(
814       cf3=-N_pl_rd,
815       createStepName='RIKS',
816       distributionType=UNIFORM,
817       field='',
818       localCsys=None,
819       name='compression',
820       region=r_model_N.rootAssembly.sets['RP-2-set']
821       )
822
823   # Field and History output requests
824
825   r_model_N.historyOutputRequests.changeKey(
826       fromName='H-Output-1',
827       toName='load'
828       )
829
830   r_model_N.historyOutputRequests['load'].setValues(
831       rebar=EXCLUDE,
832       region=r_model_N.rootAssembly.sets['RP-1-set'],
833       sectionPoints=DEFAULT, variables=('RF3', )
834       )
835
836   r_model_N.HistoryOutputRequest(
837       createStepName='RIKS',
838       name='disp',
839       rebar=EXCLUDE,
840       region=r_model_N.rootAssembly.sets['RP-2-set'],
841       sectionPoints=DEFAULT,
842       variables=('U3', )
843       )
844
845   r_model_N.HistoryOutputRequest(
846       createStepName='RIKS',
847       name='moment',
848       rebar=EXCLUDE,
849       region=r_model_N.rootAssembly.sets['RP-Mid-set'],
850       sectionPoints=DEFAULT,
851       variables=('UR1', )
852       )
853
854   r_model_N.fieldOutputRequests.changeKey(
855       fromName='F-Output-1',
856       toName='fields'
857       )
858   r_model_N.fieldOutputRequests['fields'].setValues(
859       variables=('S', 'MISES', 'E', 'PEEQ', 'U')
860       )
861
862   # Delete keyword nodefile
863   r_model_N.keywordBlock.synchVersions(storeNodesAndElements=False)
864   r_model_N.keywordBlock.replace(GetBlockPosition(r_model_N,'*End Step')-1,
      '\n')
865
866   # Change keywords to include initial imperfections file (filename was
      given wrong initially and corrected later)
867   amp_impf = s/2000
868   #r_model_N.keywordBlock.synchVersions(storeNodesAndElements=False)
869   r_model_N.keywordBlock.replace(GetBlockPosition(r_model_N, '*step')-1,
870   '\n** ----------------------------------------------------------------\n**
      \n**********GEOMETRICAL IMPERFECTIONS\n*IMPERFECTION,FILE='
```

```
871   + str(buckle_model) +',STEP=1\n1,'+ str(float(amp_impf)) +'\n2,'+
      str(float(amp_impf)) +'\n3,'+ str(float(amp_impf)) +'\n4,'+
      str(float(amp_impf)) +'\n**')
872
873   # Create Job
874   mdb.Job(
875       atTime=None,
876       contactPrint=OFF,
877       description='',
878       echoPrint=OFF,
879       explicitPrecision=SINGLE,
880       getMemoryFromAnalysis=True,
881       historyPrint=OFF,
882       memory=90,
883       memoryUnits=PERCENTAGE,
884       model='RIKS-N-1-1-1-1',
885       modelPrint=OFF,
886       multiprocessingMode=DEFAULT,
887       name='RIKS-N-1-1-1-1',
888       nodalOutputPrecision=SINGLE,
889       numCpus=1,
890       numGPUs=0,
891       queue=None,
892       resultsFormat=ODB,
893       scratch='',
894       type=ANALYSIS,
895       userSubroutine='',
896       waitHours=0,
897       waitMinutes=0
898       )
899
900   # Write the input file
901   mdb.jobs[riks_model_N].writeInput()
902
903   # RIKS model, Axial snd bending
      ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      ++++++++++++++++++++++++++++
904
905   riks_model_NM = 'RIKS-NM-'+str(i+1)+'-'+str(j+1)+'-'+str(k+1)+'-'+str(l+1)
906
907   # Copy model from buckling analysis
908   r_model_NM=mdb.Model(
909       name=riks_model_NM,
910       objectToCopy=r_model_N
911       )
912
913   # Apply bending moment at the mid-connection
914   # Calculate the magnitude of moment as 10% of moment resistance
915   W = current_Iy/(current_d/2)
916   M_resist = W*current_fy
917   M = 0.1*M_resist
918
919   r_model_NM.Moment(
920       cm1=-M,
921       createStepName='RIKS',
922       distributionType=UNIFORM,
923       field='',
924       localCsys=None,
925       name='moment',
926       region=c_assembly.sets['RP-Mid-set']
927       )
928
```

```
929   # Create Job
930   mdb.Job(
931       atTime=None,
932       contactPrint=OFF,
933       description='',
934       echoPrint=OFF,
935       explicitPrecision=SINGLE,
936       getMemoryFromAnalysis=True,
937       historyPrint=OFF,
938       memory=90,
939       memoryUnits=PERCENTAGE,
940       model='RIKS-NM-1-1-1-1',
941       modelPrint=OFF,
942       multiprocessingMode=DEFAULT,
943       name='RIKS-NM-1-1-1-1',
944       nodalOutputPrecision=SINGLE,
945       numCpus=1,
946       numGPUs=0,
947       queue=None,
948       resultsFormat=ODB,
949       scratch='',
950       type=ANALYSIS,
951       userSubroutine='',
952       waitHours=0,
953       waitMinutes=0
954       )
955
956   # Write the input file
957   mdb.jobs[riks_model_NM].writeInput()
958
959
960   # Save the model
      ------------------------------------------------------------------------
      ---------------------------
961   mdb.saveAs(pathName=os.getcwd()+'\\'+str(i+1)+'-'+str(j+1)+'-'+str(k+1)+'-'+
      str(l+1)+'.cae')
962
963
```

```
import numpy as np
import os
import string
import sys
import odbAccess
from odbAccess import *
from abaqusConstants import *
# Import pickle to load the .pkl database
import pickle

# Open and read the database
profiles_file = open("profiles.pkl",'rb')
profiles = pickle.load(profiles_file)
profiles_file.close()

profiles_file = open("meta13.pkl",'rb')
profiles_meta = pickle.load(profiles_file)
profiles_file.close()

NameOfFile='maxforcedispldata-N.txt'
out = open(NameOfFile,'w')

for b in (3, 4, 5):
    for l in (1, 2, 3):
        for k in (3, 5):
            for j in (2, 3, 4):
                for i in range (1,2,1):

                    # Variables holding information of the current profile

                    name='RIKS-N-1-'+str(j)+'-'+str(k)+'-'+str(l)+'-'+str(b)
                    +'.odb'
                    current_d =
                    float(profiles_meta[i-1][j-1][k-1][l-1][0][0])
                    current_t =
                    float(profiles_meta[i-1][j-1][k-1][l-1][1][0])
                    current_l =
                    float(profiles_meta[i-1][j-1][k-1][l-1][7][0])
                    current_fy =
                    float(profiles_meta[i-1][j-1][k-1][l-1][3][0])
                    current_area =
                    float(profiles_meta[i-1][j-1][k-1][l-1][4][0])
                    current_Iy =
                    float(profiles_meta[i-1][j-1][k-1][l-1][5][0])
                    current_effarealocal =
                    float(profiles_meta[i-1][j-1][k-1][l-1][8][0])
                    current_class =
                    float(profiles_meta[i-1][j-1][k-1][l-1][9][0])
                    current_It =
                    float(profiles_meta[i-1][j-1][k-1][l-1][10][0])
                    current_Iw =
                    float(profiles_meta[i-1][j-1][k-1][l-1][11][0])
                    current_corneround1 =
                    float(profiles_meta[i-1][j-1][k-1][l-1][12][0])
                    current_corneround2 =
                    float(profiles_meta[i-1][j-1][k-1][l-1][13][0])
                    current_bp =
                    float(profiles_meta[i-1][j-1][k-1][l-1][14][0])
                    current_llip =
                    float(profiles_meta[i-1][j-1][k-1][l-1][15][0])

                    E=210000
```

```
46                      G=81000
47                      epsilon=(sqrt(235/current_fy))*(sqrt(235/current_fy))
48                      current_lambda =
                        current_l/(pi*sqrt(E*current_Iy/(current_area*current_fy
                        )))
49
50                      nameOfStep='RIKS'
51                      myOdb = odbAccess.openOdb(path=name)
52                      RIKS=
                        myOdb.steps[nameOfStep]
53                      rp1key = RIKS.historyRegions.keys()[1]
54                      ho1key =
                        RIKS.historyRegions[rp1key].historyOutputs.keys()[0]
55                      rp2key = RIKS.historyRegions.keys()[2]
56                      ho2key =
                        RIKS.historyRegions[rp2key].historyOutputs.keys()[0]
57                      load_hist =
                        RIKS.historyRegions[rp1key].historyOutputs[ho1key].data
58                      disp_hist =
                        RIKS.historyRegions[rp2key].historyOutputs[ho2key].data
59                      maxpos = load_hist.index(max(load_hist,key=lambda
                        x:x[1]))
60                      load = load_hist[maxpos][1]
61                      disp = -disp_hist[maxpos][1]
62
63                      ## Calculation of elastic buckling based on EC
64                      current_Pb =
                        pi*pi*E*current_Iy/((current_l+1.5*current_d)*(current_l
                        +1.5*current_d))
65
66                      ## Calculation of average yield strength, according to
                        EC-3-1-3
67                      current_fya =
                        current_fy+(520-current_fy)*7*6*current_t*current_t/curr
                        ent_area
68                      limit_fya = (520+current_fy)/2
69                      if current_fya >= limit_fya:
70                          current_fya = limit_fya
71
72                      ## Calculation of reduction factor due to distortional
73                      # Stiffness of stiffener
74                      current_delta =
                        ((1*current_bp*current_bp*current_bp*current_bp)/(3*(cur
                        rent_bp+current_bp)))*((12*(1-0.3*0.3))/(E*current_t*cur
                        rent_t*current_t))
75                      u = 1
76                      current_K = u/current_delta
77                      # area of stiffener
78                      current_As =
                        (current_llip*2+current_bp+current_bp)*current_t

79                      # moment of inertia stiffener
80                      current_z
                        =((current_bp*2*current_t*current_t/2+current_llip*2*cur
                        rent_t*current_llip))/current_As
81                      current_Is =
                        (1/12)*current_t*current_t*current_t*2*current_bp+2*curr
                        ent_bp*current_t*(current_z-current_t/2)*(current_z-curr
                        ent_t/2)+(1/12)*2*current_llip*2*current_llip*2*current_
                        llip*current_t+2*current_llip*current_t*(current_llip-cu
                        rrent_z)*(current_llip-current_z)
82
```

```
 83                      #critical buckling stress of stiffener
 84                      current_sigmacrs =
                         2*sqrt(current_K*E*current_Is)/current_As
 85                      #lambdabar distortional
 86                      current_lambdabard = sqrt(current_fy/current_sigmacrs)
 87                      #reduction factor distotional
 88                      if current_lambdabard <= 0.65:
 89                          current_Chid = 1
 90                      else:
 91                          current_Chid = 1.47-0.723*current_lambdabard
 92
 93                      # Iteration of reduction factor for less conservative
                         result
 94                      #Iteration 1
 95                      gamma_m0 = 1.0
 96                      current_sigmacomEd_1 = current_Chid*current_fy/gamma_m0
 97                      current_lambdap_llip =
                         (current_llip*2/current_t)/(28.4*epsilon*2)
 98                      current_lambdap_1 =
                         sqrt(current_Chid)*current_lambdap_llip
 99                      current_rop_1 =
                         (current_lambdap_1-0.188)/(current_lambdap_1*current_lam
                         bdap_1)
100                      if current_rop_1 > 1:
101                          current_rop_1 = 1
102                      current_llip_1 = current_rop_1*current_llip
103
104                      current_lambdap_bp =
                         (current_bp/current_t)/(28.4*epsilon*2)
105                      current_lambdapbp_1 =
                         sqrt(current_Chid)*current_lambdap_bp
106                      current_ropbp_1 =
                         (current_lambdapbp_1-0.055*4)/(current_lambdapbp_1*curre
                         nt_lambdapbp_1)
107                      if current_ropbp_1 > 1:
108                          current_ropbp_1 = 1
109                      current_bp_1 =
                         current_ropbp_1*current_bp
110                      current_As_1 =
                         (current_llip_1*2+current_bp_1+current_bp_1)*current_t
111                      current_z_1
                         =((current_bp_1*2*current_t*current_t/2+current_llip_1*2
                         *current_t*current_llip_1))/current_As_1
112                      current_Is_1=
                         (1/12)*current_t*current_t*current_t*2*current_bp_1+2*cu
                         rrent_bp_1*current_t*(current_z-current_t/2)*(current_z_
                         1-current_t/2)+(1/12)*2*current_llip_1*2*current_llip_1*
                         2*current_llip_1*current_t+2*current_llip_1*current_t*(c
                         urrent_llip_1-current_z_1)*(current_llip_1-current_z_1)
113                      current_sigmacrs_1 =
                         2*sqrt(current_K*E*current_Is_1)/current_As_1
114                      current_lambdabard_1 =
                         sqrt(current_fy/current_sigmacrs_1)
115                      if current_lambdabard_1 <= 0.65:
116                          current_Chid_1 = 1
117                      else:
118                          current_Chid_1 =
                             1.47-0.723*current_lambdabard_1
119
120                      #Iteration 2
121                      current_sigmacomEd_2 =
                         current_Chid_1*current_fy/gamma_m0
```

```
122                     current_lambdap_2 =
                        sqrt(current_Chid_1)*current_lambdap_llip
123                     current_rop_2 =
                        (current_lambdap_2-0.188)/(current_lambdap_2*current_lam
                        bdap_2)
124                     if current_rop_2 > 1:
125                         current_rop_2 = 1
126                     current_llip_2 =
                        current_rop_2*current_llip
127                     current_lambdapbp_2 =
                        sqrt(current_Chid_1)*current_lambdap_bp
128                     current_ropbp_2 =
                        (current_lambdapbp_2-0.055*4)/(current_lambdapbp_2*curre
                        nt_lambdapbp_2)
129                     if current_ropbp_2 > 1:
130                         current_ropbp_2 = 1
131                     current_bp_2 =
                        current_ropbp_2*current_bp
132                     current_As_2 =
                        (current_llip_2*2+current_bp_2+current_bp_2)*current_t
133                     current_z_2
                        =((current_bp_2*2*current_t*current_t/2+current_llip_2*2
                        *current_t*current_llip_2))/current_As_2
134                     current_Is_2=
                        (1/12)*current_t*current_t*current_t*2*current_bp_2+2*cu
                        rrent_bp_2*current_t*(current_z_2-current_t/2)*(current_
                        z_2-current_t/2)+(1/12)*2*current_llip_2*2*current_llip_
                        2*2*current_llip_2*current_t+2*current_llip_2*current_t*
                        (current_llip_2-current_z_2)*(current_llip_2-current_z_2
                        )
135                     current_sigmacrs_2 =
                        2*sqrt(current_K*E*current_Is_2)/current_As_2
136                     current_lambdabard_2 =
                        sqrt(current_fy/current_sigmacrs_2)
137                     if current_lambdabard_2 <= 0.65:
138                         current_Chid_2 = 1
139                     else:
140                         current_Chid_2 = 1.47-0.723*current_lambdabard_2
141
142                     ## Effective area due to distortional
143                     current_effareadistort =
                        current_effarealocal-((1-current_Chid_2)*current_t*6*(cu
                        rrent_llip+current_bp))
144
145                     # Calculation of resistance according to EC3-1-3 Part 6
146                     # Cross section resistance pure axial compression
147                     gamma_m0 = 1.0
148                     current_Ncrd_local = current_area*current_fy/gamma_m0
149                     current_Ncrd_local_fya =
                        current_area*current_fya/gamma_m0
150                     current_Ncrd =
                        current_effareadistort*current_fy/gamma_m0
151                     current_Ncrd_fya =
                        current_effareadistort*current_fya/gamma_m0
152
153                     # Cross section resistance pure bending moment
154                     current_W = current_Iy/(current_d/2)
155                     current_Mcrd = current_W*current_fy/gamma_m0
156
157                     #Buckling resistance
158                     #Flexural buckling
159                     alpha=0.34  #buckling coefficient
```

```
160                     current_lcr = current_l
161                     current_i = sqrt(current_Iy/current_area)   #radius of
                        gyration
162                     current_sigmacrit =
                        ((pi*pi)*E)/((current_lcr/current_i)*(current_lcr/curren
                        t_i))
163                     current_lambdabar =
                        sqrt((current_fy*current_effareadistort)/(current_sigmac
                        rit*current_area))
164                     current_Phi = 0.5*(1+alpha*(current_lambdabar -
                        0.2)+(current_lambdabar*current_lambdabar))
165                     current_Chi =
                        1/(current_Phi+sqrt((current_Phi*current_Phi)-(current_l
                        ambdabar*current_lambdabar)))
166                     if current_Chi>1:
167                         current_Chi = 1
168                     current_Nbf =
                        current_Chi*current_effareadistort*current_fy
169
170                     #Torsional buckling
171                     current_io =
                        sqrt((current_i*current_i)+(current_i*current_i))
172                     current_sigmacritT =
                        (1/(current_area*(current_io*current_io)))*(G*current_It
                        +((pi*pi*E*current_Iw)/(current_lcr*current_lcr)))
173                     current_lambdabarT =
                        sqrt((current_fy*current_effareadistort)/(current_sigmac
                        ritT*current_area))
174                     current_PhiT = 0.5*(1+alpha*(current_lambdabarT -
                        0.2)+(current_lambdabarT*current_lambdabarT))
175                     current_ChiT =
                        1/(current_PhiT+sqrt((current_PhiT*current_PhiT)-(curren
                        t_lambdabarT*current_lambdabarT)))
176                     if current_ChiT>1:
177                         current_ChiT = 1
178                     current_NbT =
                        current_ChiT*current_effareadistort*current_fy

179
180                     #Flexural-Torsional buckling
181                     beta = 1.0
182                     current_sigmacritFT =
                        (current_sigmacrit/(2*beta))*(1+(current_sigmacritT/curr
                        ent_sigmacrit)-(sqrt((1-(current_sigmacritT/current_sigm
                        acrit))*(1-(current_sigmacritT/current_sigmacrit)))+(4*0
                        *current_sigmacritT/current_sigmacrit)))
183                     current_lambdabarFT =
                        sqrt((current_fy*current_effareadistort)/(current_sigmac
                        ritFT*current_area))
184                     current_PhiFT = 0.5*(1+alpha*(current_lambdabarFT -
                        0.2)+(current_lambdabarFT*current_lambdabarFT))
185                     current_ChiFT =
                        1/(current_PhiFT+sqrt((current_PhiFT*current_PhiFT)-(cur
                        rent_lambdabarFT*current_lambdabarFT)))
186                     if current_ChiFT>1:
187                         current_ChiFT = 1
188                     current_NbFT =
                        current_ChiFT*current_effareadistort*current_fy

189
190                     #Final buckling
191                     current_Nbrd =
```

```
                        current_effareadistort*current_fy*min(current_Chi,
                        current_ChiT, current_ChiFT)
192                     current_Nbrd_fya =
                        current_effareadistort*current_fya*min(current_Chi,
                        current_ChiT, current_ChiFT)
193
194                     ## Calculation of resistance according to EC3-1-6 Shell
195                     Cxb = 1  #pinned-pinned BC
196                     Q = 16  #fabrication class C
197                     lambdaxo = 0.2
198                     betax = 0.6
199                     gammam1 = 1.1
200                     current_w = current_l/sqrt((current_d/2)/current_t)
201                     limit = 0.5*(current_d/2)/current_t
202                     if current_w>limit:
203                         current_Cx =
                            1+((0.2/Cxb)*(1-2*current_w*current_t/(current_d/2))
                            )
204                         if current_Cx<0.6:
205                             current_Cx=0.6
206                         current_sigmaxRcr =
                            0.605*E*current_Cx*current_t/(current_d/2)
                            #critical meridional buckling stress
207                         current_alphax =
                            0.62/(1+(1.91*((1/Q)*sqrt(current_d/2)/current_t)*1.
                            44))
208                         current_lambdabarp=sqrt(current_alphax/(1-betax))
209                         current_lambdax = sqrt(current_fy/current_sigmaxRcr)
210                         current_Chix =
                            1-betax*((current_lambdax-lambdaxo)/(current_lambdab
                            arp-lambdaxo))
211                         current_sigmaRd =
                            current_Chix*current_fy/gammam1       #design
                            resistance
212                         current_Ncrdshell = current_sigmaRd*current_area
213
214                     ## Calculation of weight of the member
215                     gamma_steel = 7850
216                     current_weight =
                        current_area*(current_l*2+current_d*3)*gamma_steel*0.000
                        000001
217
218
                        out.write(str(i)+'\t'+str(j)+'\t'+str(j)+'\t'+str(k)+'\t
                        '+str(l)+'\t'+str(b)+'\t'+str(current_d)+'\t'+str(curren
                        t_t)+'\t'+str(current_l*2+current_d*3)+'\t'+str(round(cu
                        rrent_d/current_t,2))+'\t'+str(current_lambda)+'\t'+str(
                        b)+'\t'+str(load)+'\t'+
219
                        str(disp)+'\t'+str(current_d/(current_t*epsilon))+'\t'+s
                        tr(current_area)+'\t'+str(current_effarealocal)+'\t'+str
                        (current_effareadistort)+'\t'+str(current_class)+'\t'+st
                        r(current_Ncrd_local)+'\t'+str(current_Ncrd)+'\t'+str(cu
                        rrent_Ncrd_local_fya)+'\t'+str(current_Ncrd_fya)+'\t'+st
                        r(current_Chi)+'\t'+str(round(current_ChiT,0))+'\t'+str(
                        current_ChiFT)+'\t'+str(current_Nbrd)+'\t'+str(current_N
                        brd_fya)+'\t'+str(current_Ncrdshell)+'\t'+str(current_fy
                        a)+'\t'+str(current_Chid_2)+'\t'+str(current_Pb)+'\t'+st
                        r(current_weight)+'\t'+str((load/1000)/current_weight)+'
                        \n')
220                     myOdb.close()
221   out.close()
```

ANNEX B.1. LOAD-DISPLACEMENT CURVES – RIKS N

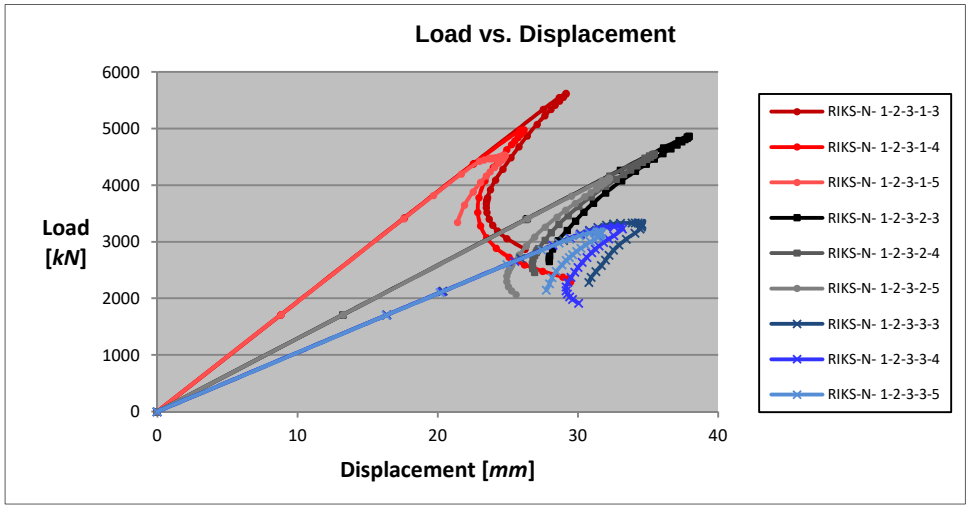


Figure B.1 Load-displacement curve for models with kept d=500, slend=90; varied lambda λ=0.65, 1.0 & 1.25, bolt ratio b=3, 4, & 5

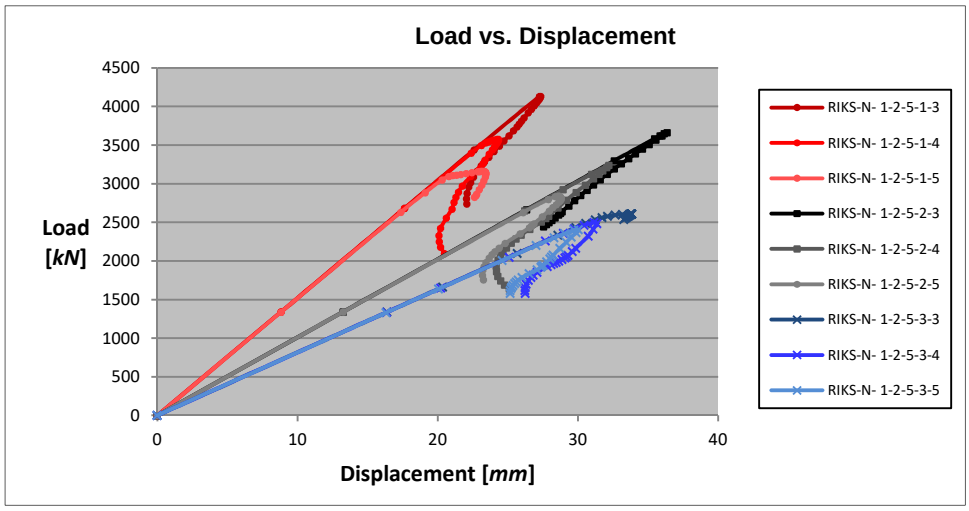


Figure B.2 Load-displacement curve for models with kept d=500, slend=110; varied lambda λ=0.65, 1.0 & 1.25, bolt ratio b=3, 4, & 5

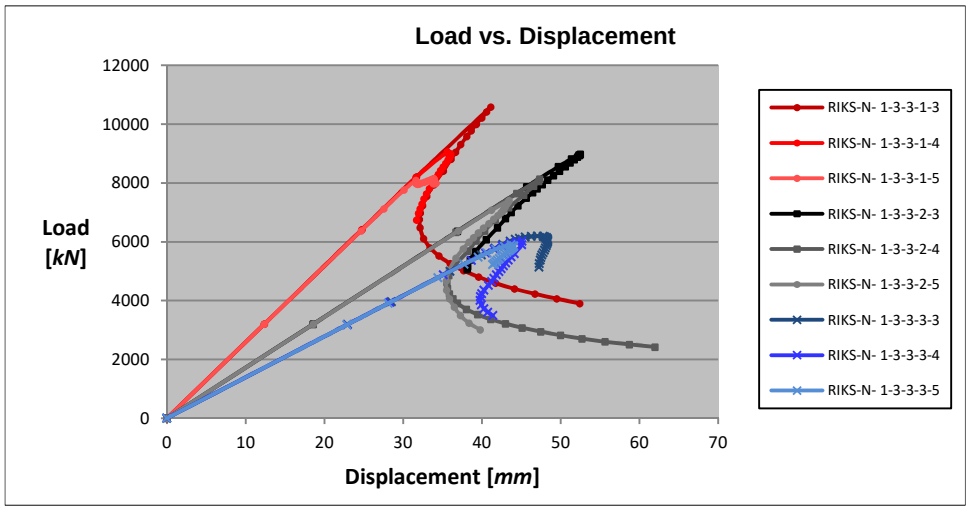


Figure B.3 Load-displacement curve for models with kept d=700, slend=90; varied lambda λ=0.65, 1.0 & 1.25, bolt ratio b=3, 4, & 5

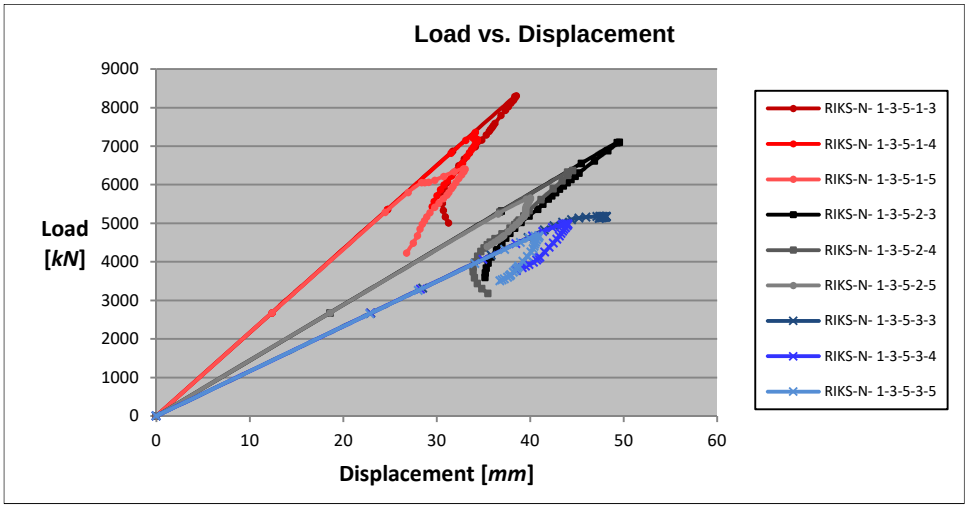


Figure B.4 Load-displacement curve for models with kept d=700, slend=110; varied lambda λ=0.65, 1.0 & 1.25, bolt ratio b=3, 4, & 5

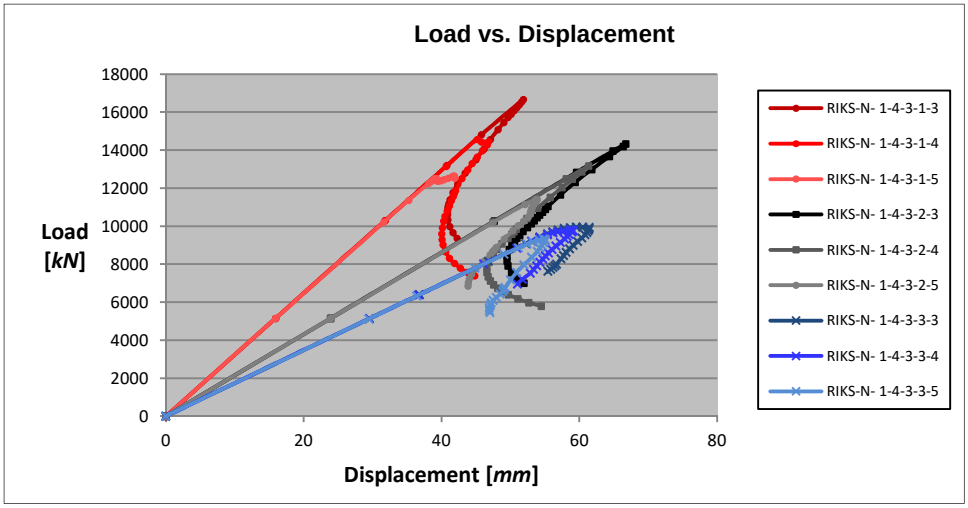


Figure B.5 Load-displacement curve for models with kept d=900, slend=90; varied lambda λ=0.65, 1.0 & 1.25, bolt ratio b=3, 4, & 5

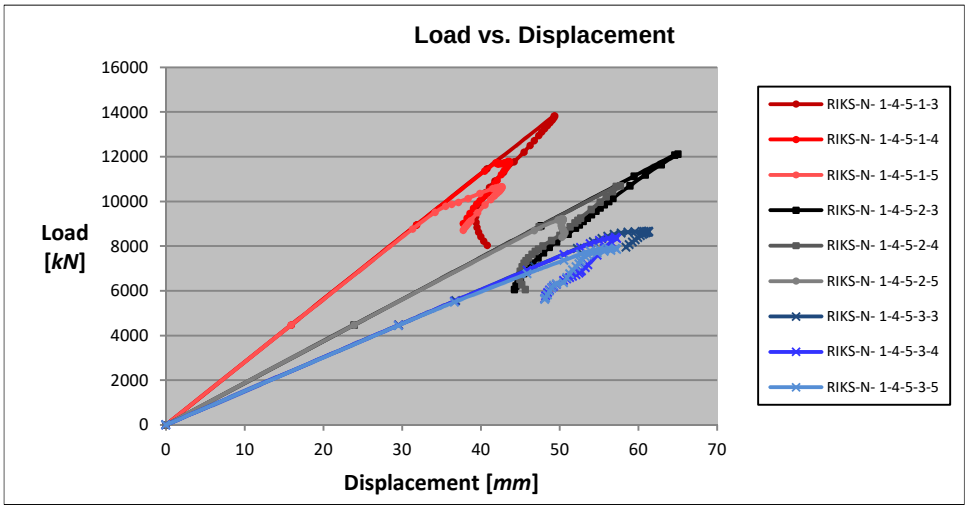


Figure B.6 Load-displacement curve for models with kept d=900, slend=110; varied lambda λ=0.65, 1.0 & 1.25, bolt ratio b=3, 4, & 5

ANNEX B.2. LOAD-ROTATION CURVES – RIKS NM

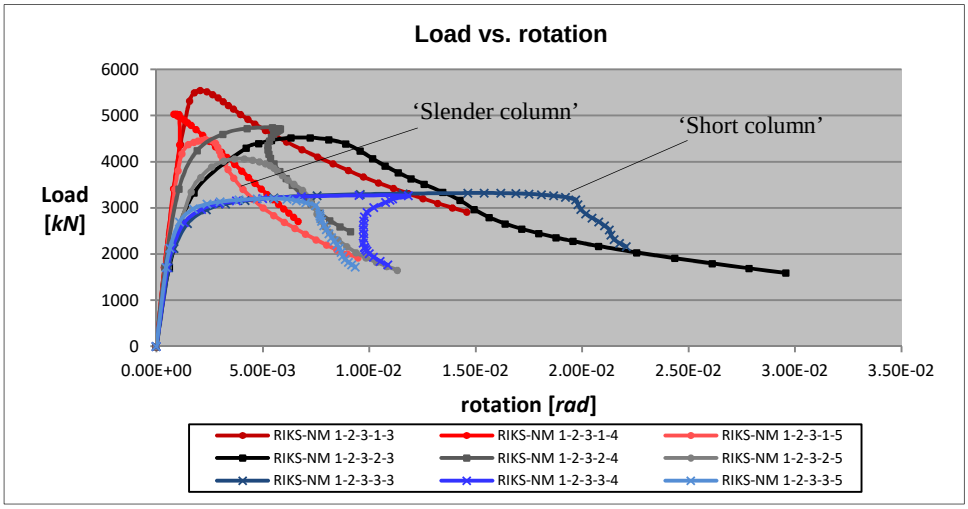


Figure B.1 Load-rotation curve for models with kept d=500, slend=90; varied lambda λ=0.65, 1.0 & 1.25, bolt ratio b=3, 4, & 5

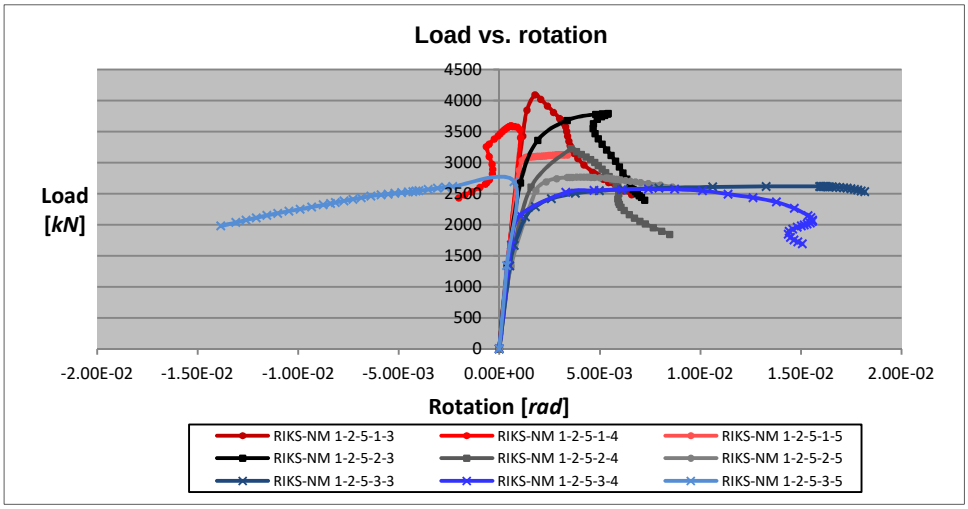


Figure B.2 Load-rotation curve for models with kept d=500, slend=110; varied lambda λ=0.65, 1.0 & 1.25, bolt ratio b=3, 4, & 5

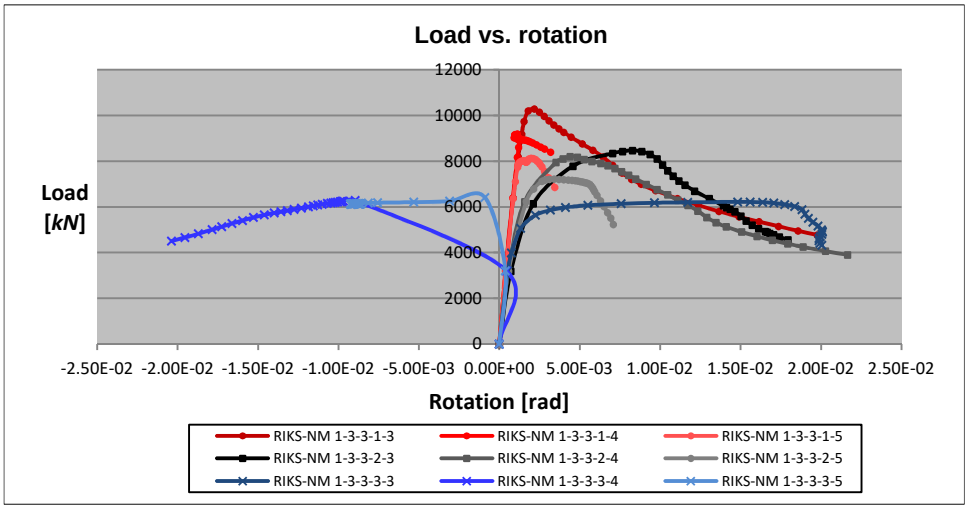


Figure B.3 Load-rotation curve for models with kept d=700, slend=90; varied lambda λ=0.65, 1.0 & 1.25, bolt ratio b=3, 4, & 5

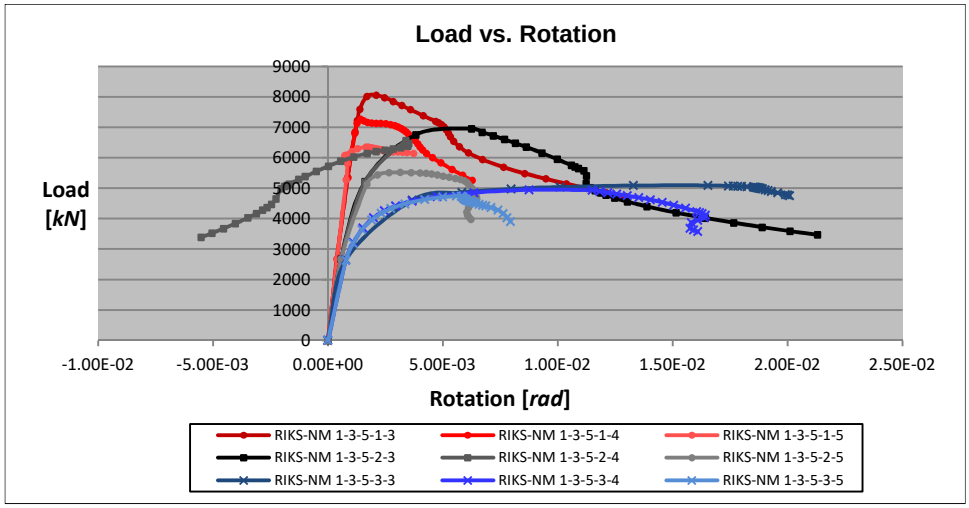


Figure B.4 Load-rotation curve for models with kept d=700, slend=110; varied lambda λ=0.65, 1.0 & 1.25, bolt ratio b=3, 4, & 5

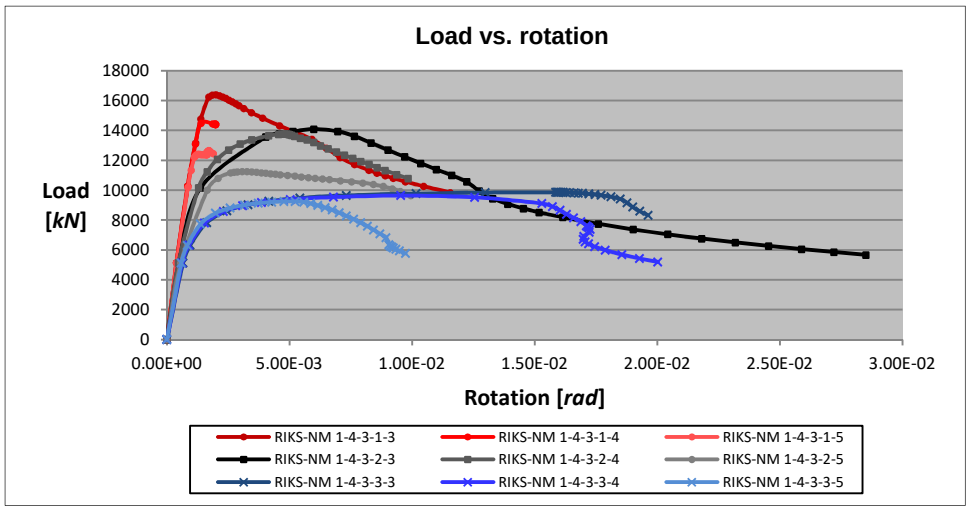


Figure B.5 Load-rotation curve for models with kept d=900, slend=90; varied lambda λ=0.65, 1.0 & 1.25, bolt ratio b=3, 4, & 5

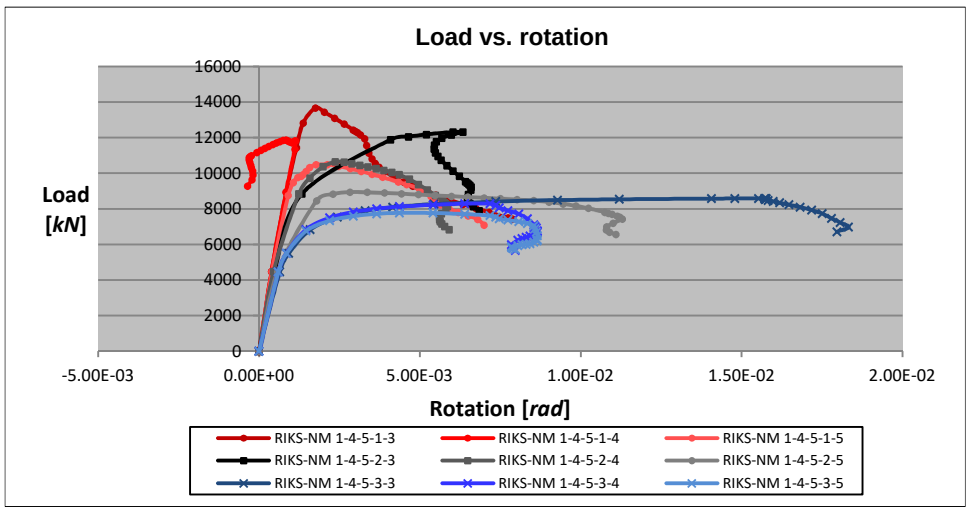


Figure B.6 Load-rotation curve for models with kept d=900, slend=110; varied lambda λ=0.65, 1.0 & 1.25, bolt ratio b=3, 4, & 5

ANNEX B.3. AXIAL -BENDING MOMENT N-M INTERACTION

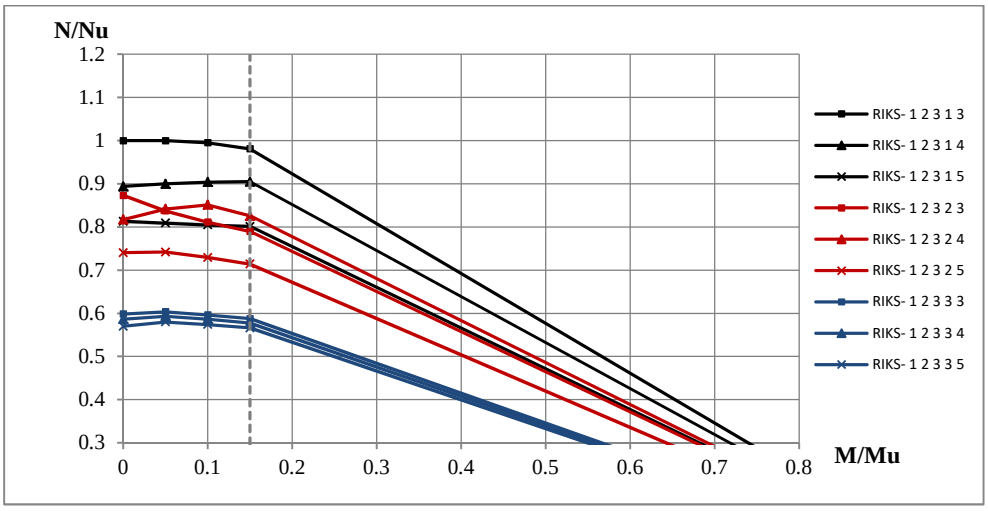


Figure B.1 N-M interaction curve for models with kept d=500, slend=90; varied lambda λ=0.65, 1.0 & 1.25, bolt ratio b=3, 4, & 5

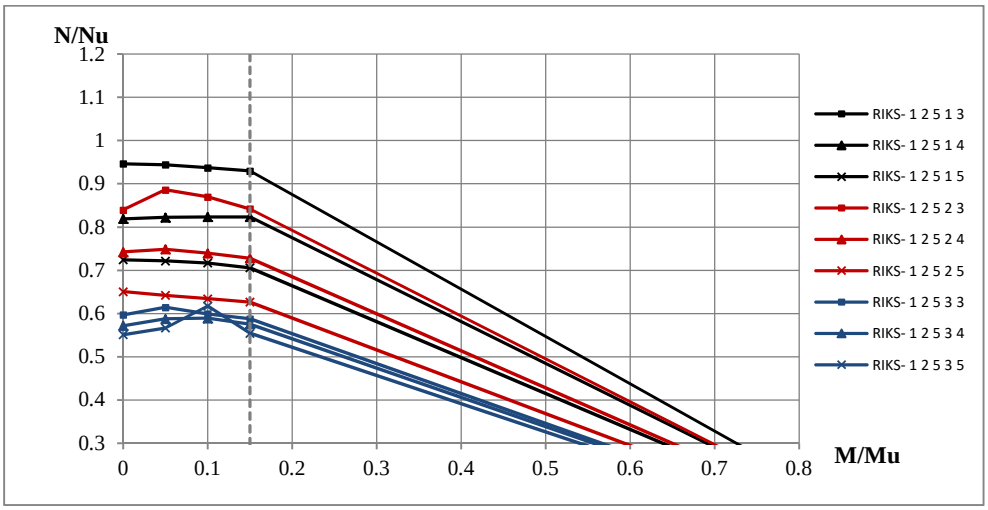


Figure B.2 N-M interaction curve for models with kept d=500, slend=110; varied lambda λ=0.65, 1.0 & 1.25, bolt ratio b=3, 4, & 5

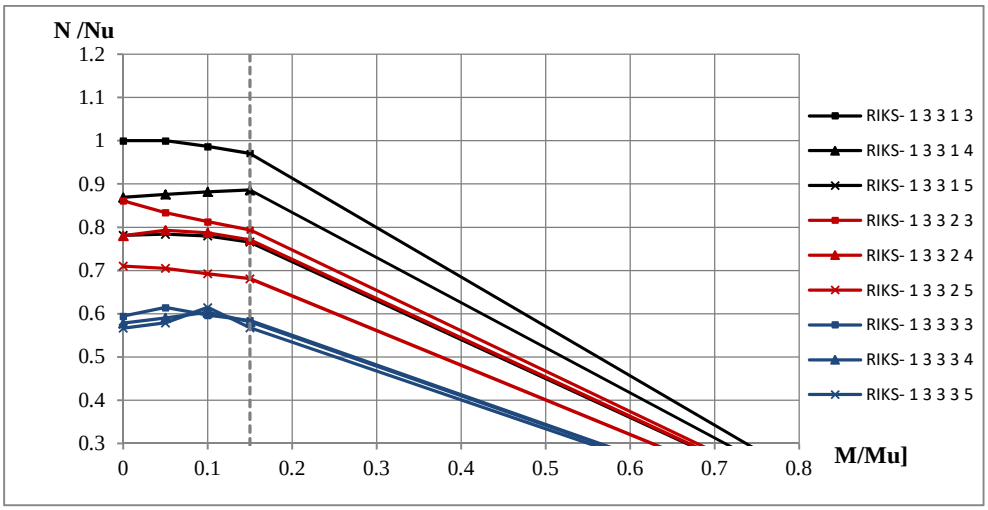


Figure B.3 N-M interaction curve for models with kept d=700, slend=90; varied lambda λ=0.65, 1.0 & 1.25, bolt ratio b=3, 4, & 5

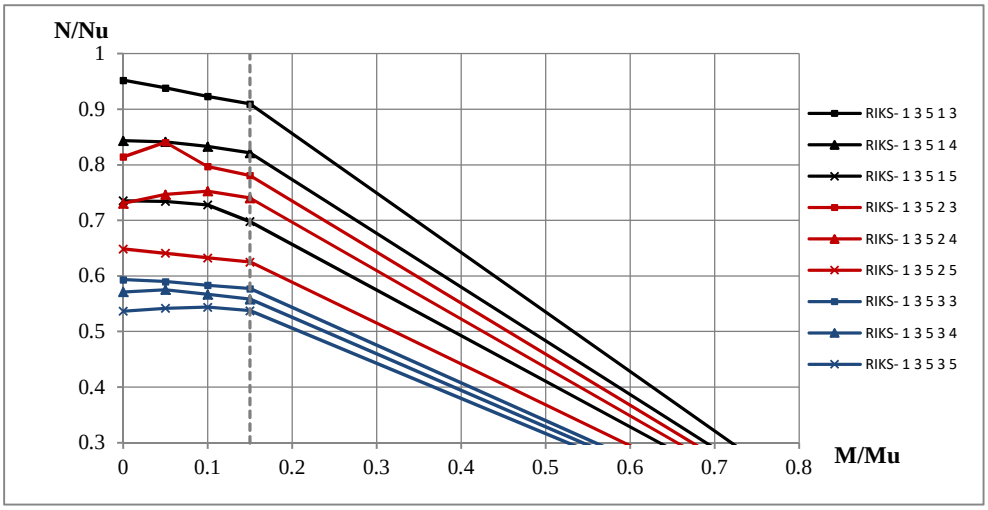


Figure B.4 N-M interaction curve for models with kept d=700, slend=110; varied lambda λ=0.65, 1.0 & 1.25, bolt ratio b=3, 4, & 5

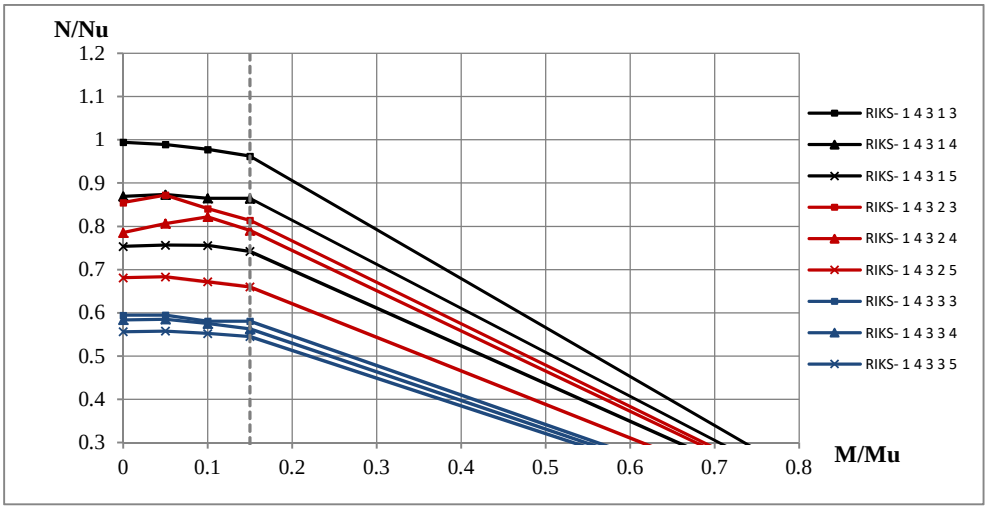


Figure B.5 N-M interaction curve for models with kept d=900, slend=90; varied lambda λ=0.65, 1.0 & 1.25, bolt ratio b=3, 4, & 5

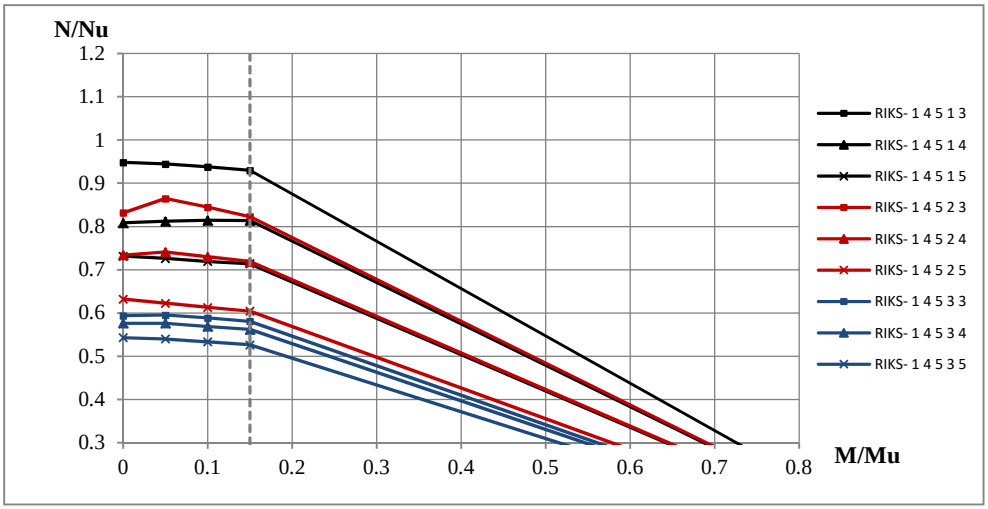
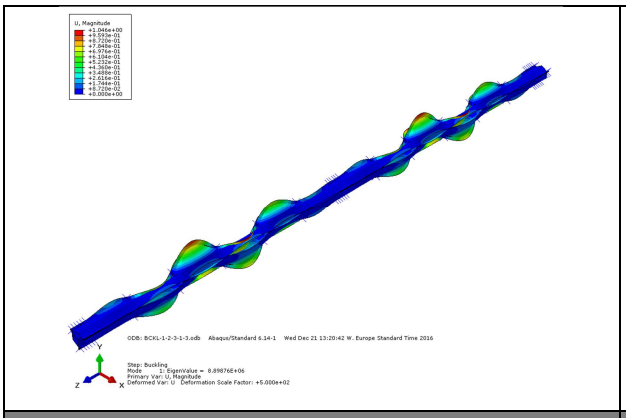
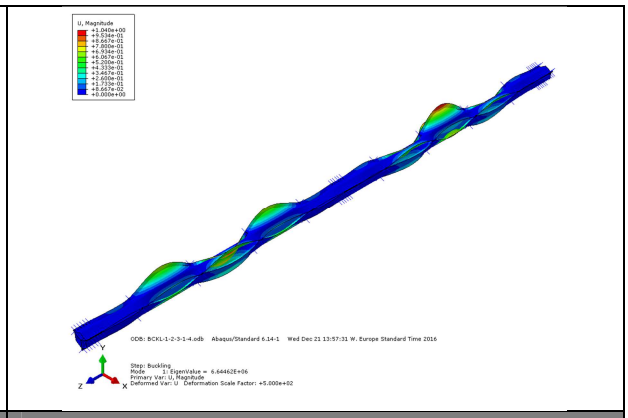


Figure B.6 N-M interaction curve for models with kept d=900, slend=110; varied lambda λ=0.65, 1.0 & 1.25, bolt ratio b=3, 4, & 5

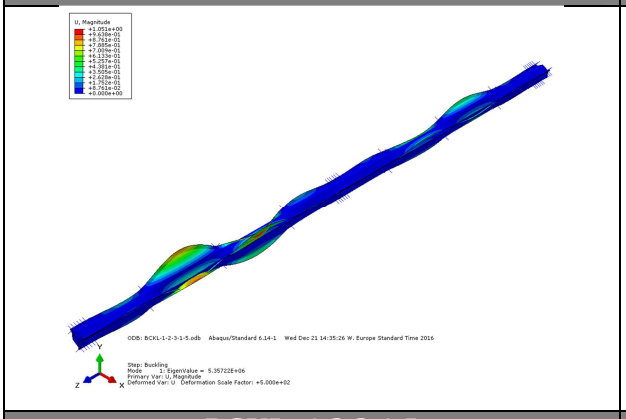
ANNEX C.1. ELASTIC BUCKLING EIGENMODES



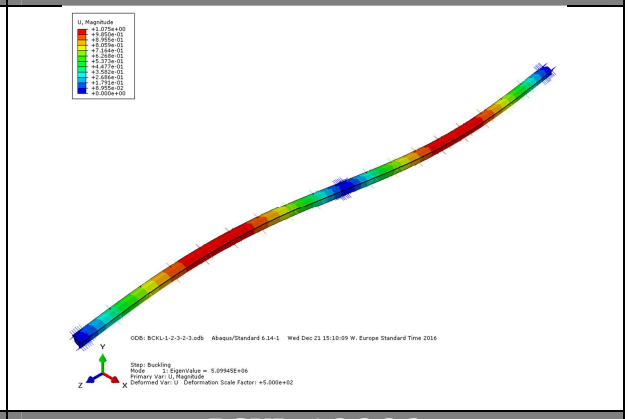
BCKL- 1-2-3-1-3



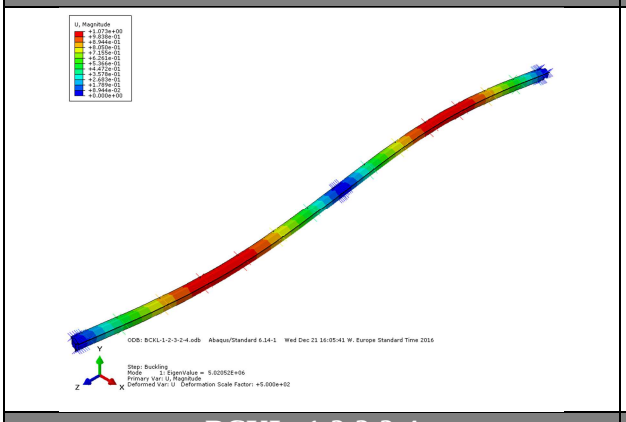
BCKL- 1-2-3-1-4



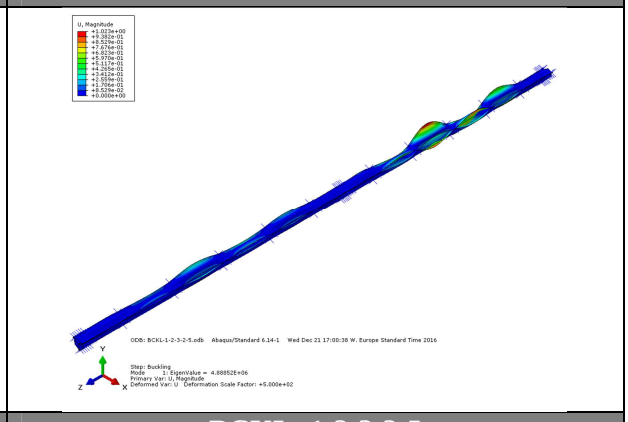
BCKL- 1-2-3-1-5



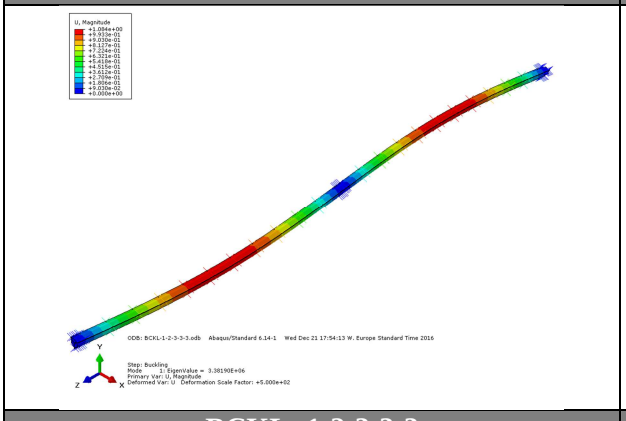
BCKL- 1-2-3-2-3



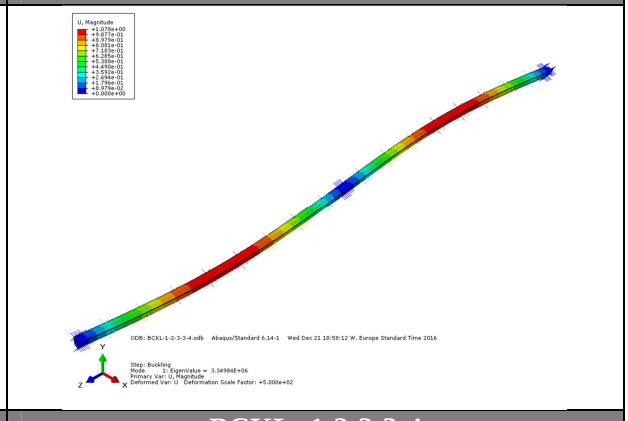
BCKL- 1-2-3-2-4



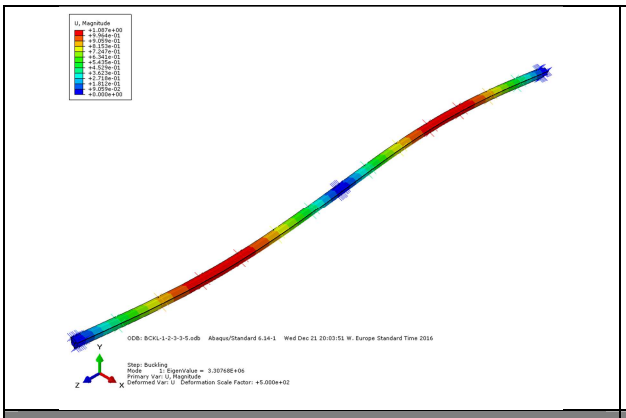
BCKL- 1-2-3-2-5



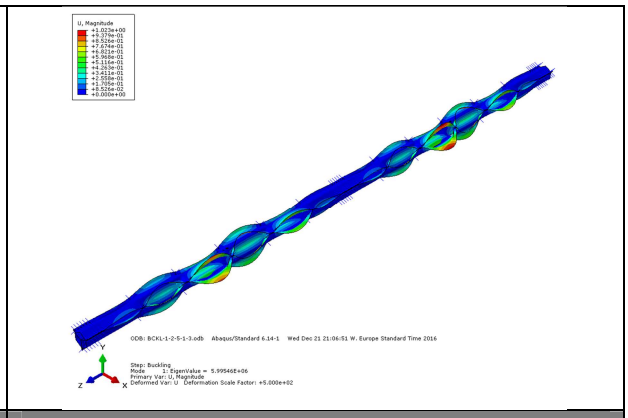
BCKL- 1-2-3-3-3



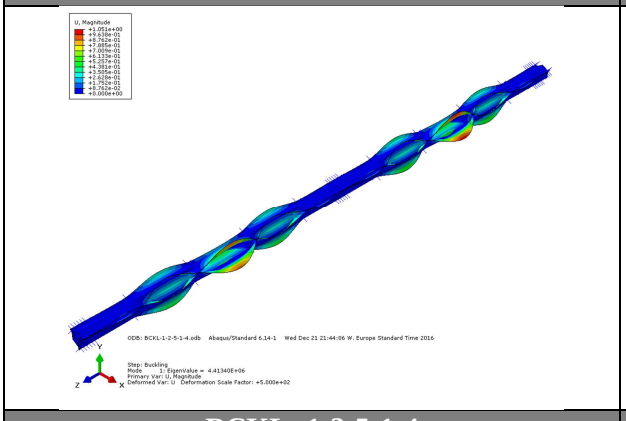
BCKL- 1-2-3-3-4



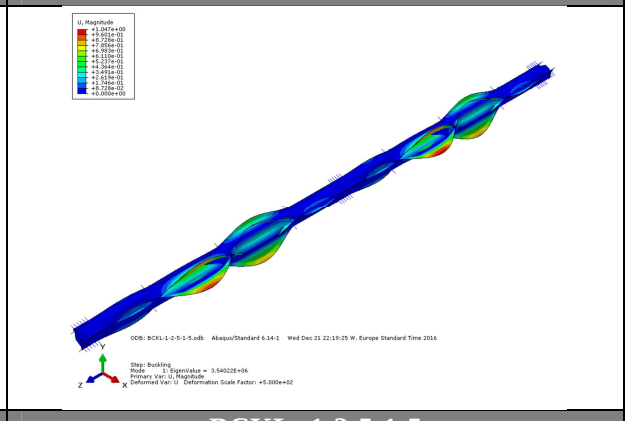
BCKL- 1-2-3-3-5



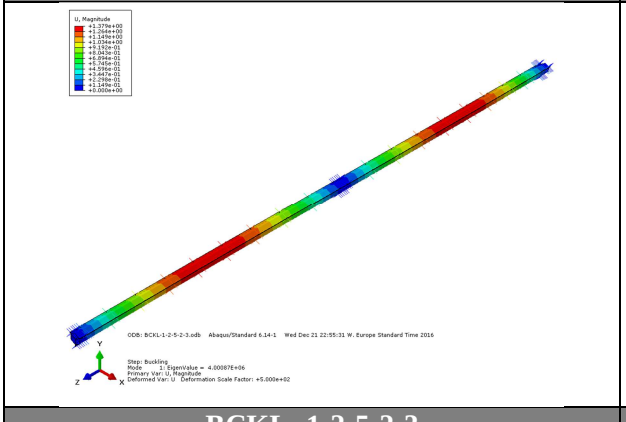
BCKL- 1-2-5-1-3



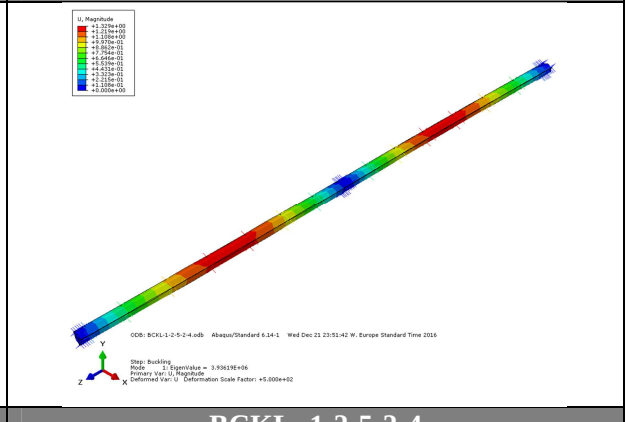
BCKL- 1-2-5-1-4



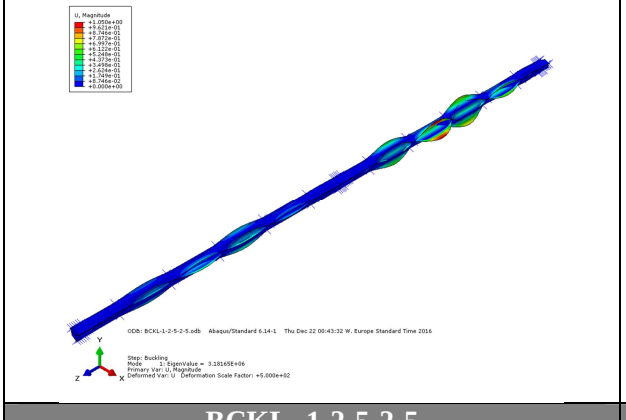
BCKL- 1-2-5-1-5



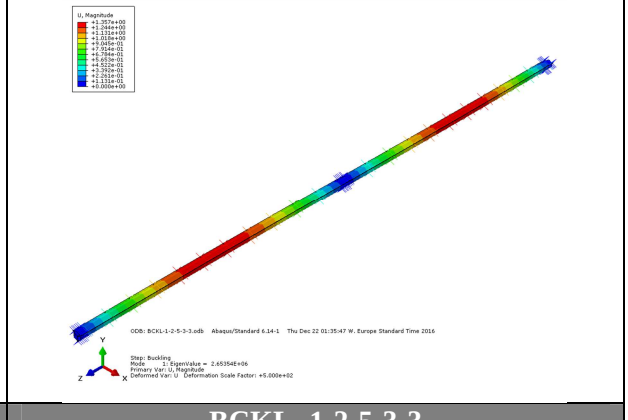
BCKL- 1-2-5-2-3



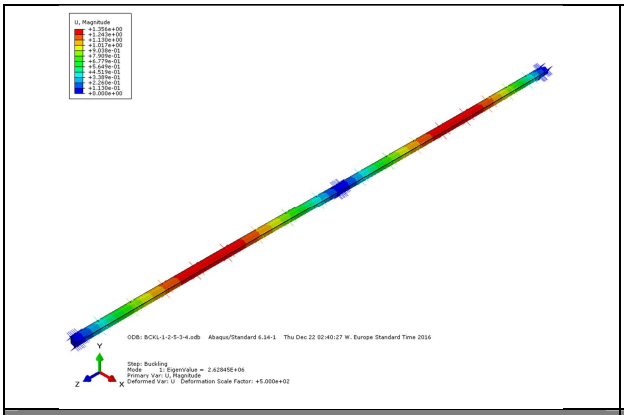
BCKL- 1-2-5-2-4



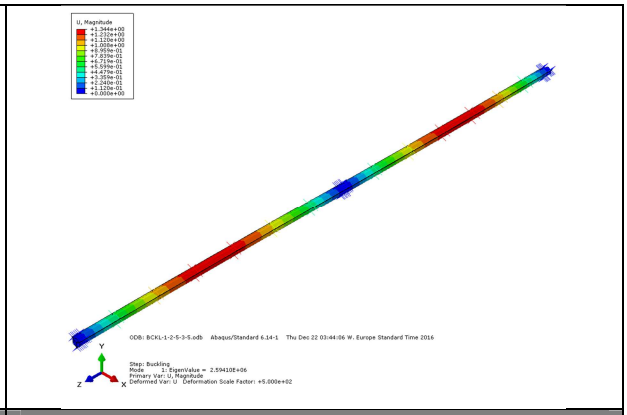
BCKL- 1-2-5-2-5



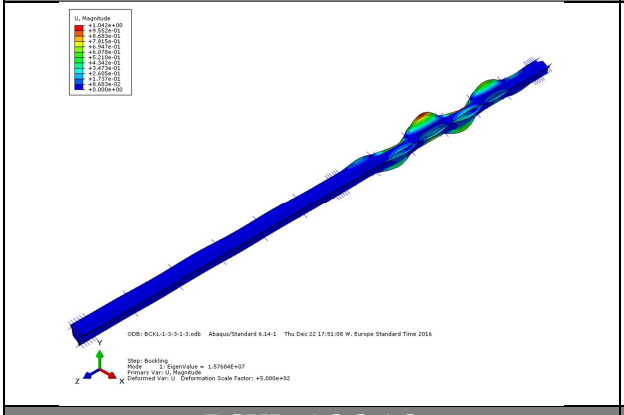
BCKL- 1-2-5-3-3



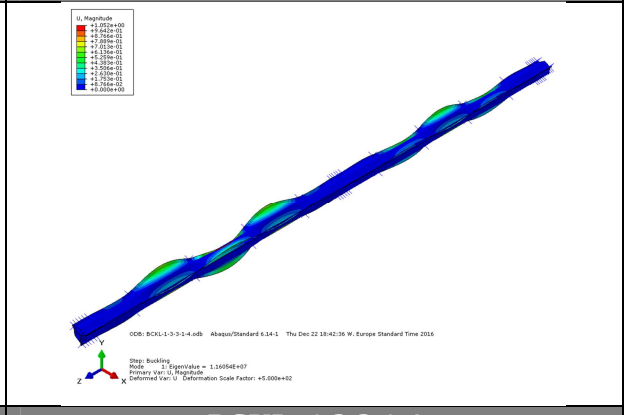
BCKL- 1-2-5-3-4



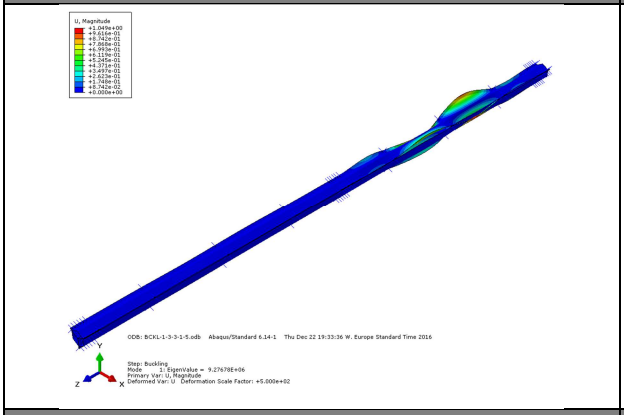
BCKL- 1-2-5-3-5



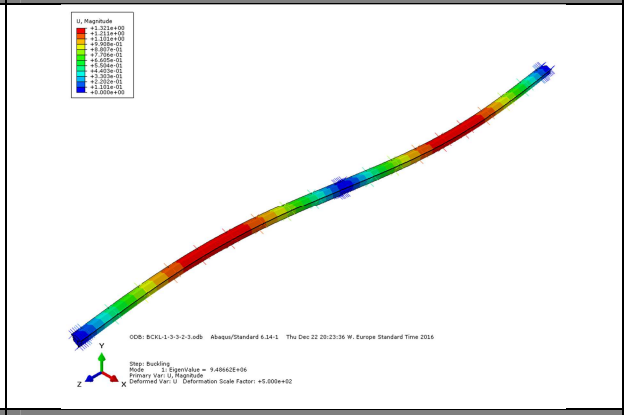
BCKL- 1-3-3-1-3



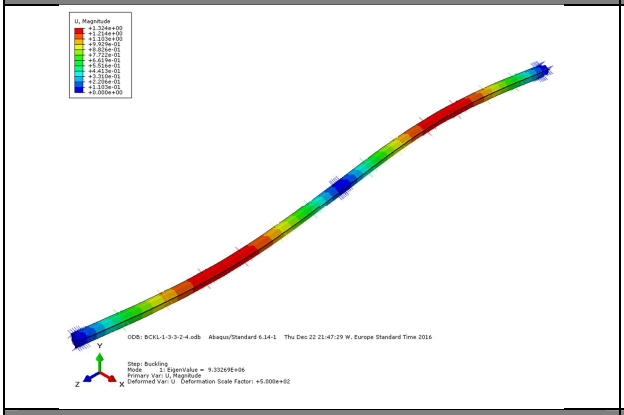
BCKL- 1-3-3-1-4



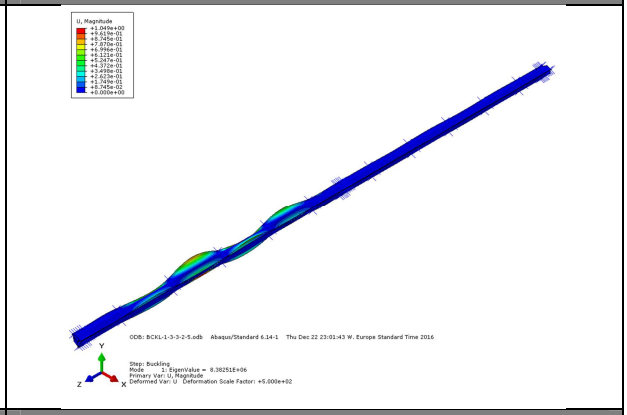
BCKL- 1-3-3-1-5



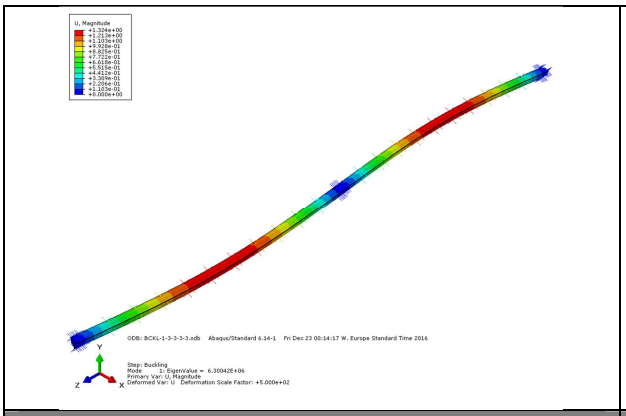
BCKL- 1-3-3-2-3



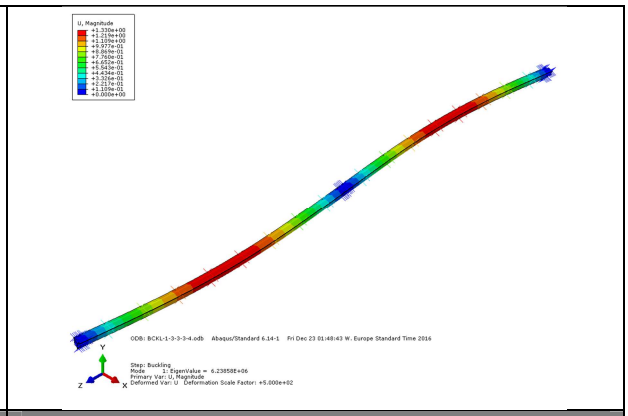
BCKL- 1-3-3-2-4



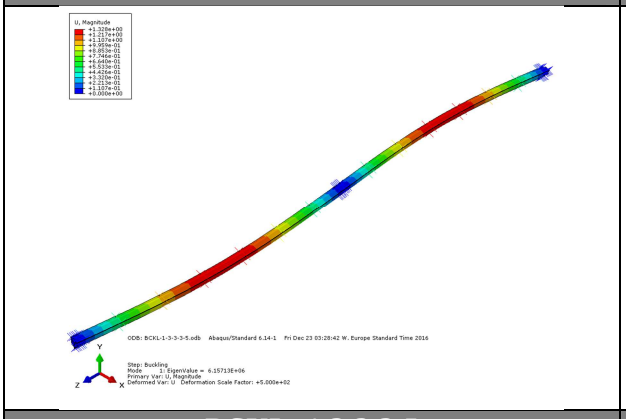
BCKL- 1-3-3-2-5



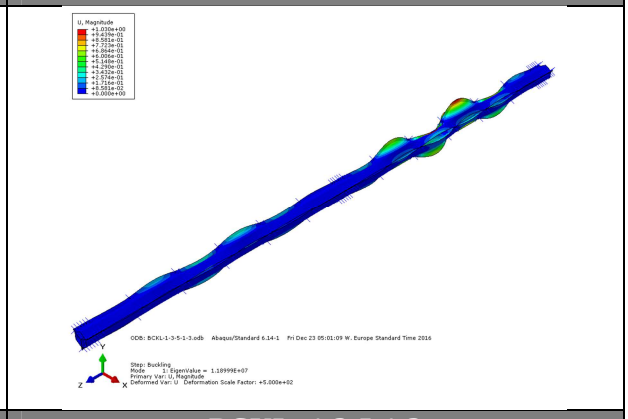
BCKL- 1-3-3-3-3



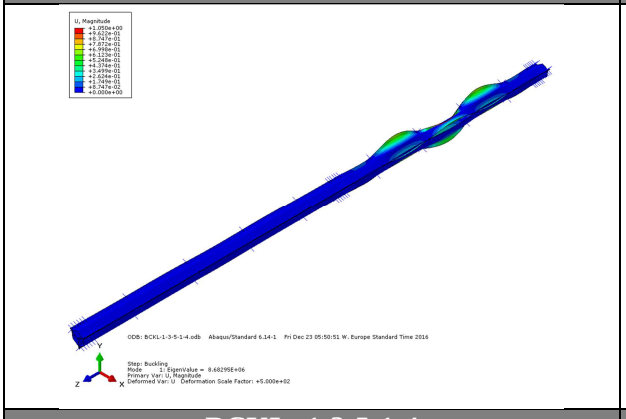
BCKL- 1-3-3-3-4



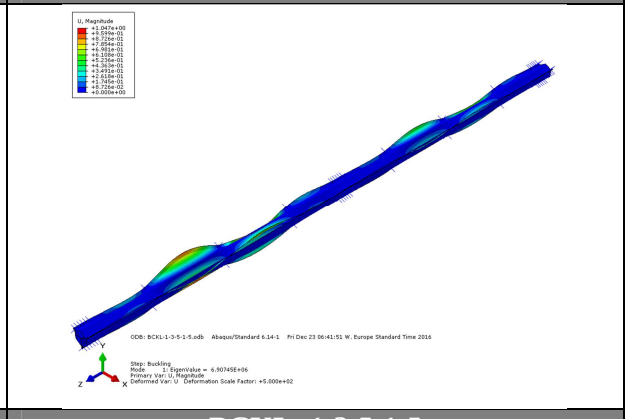
BCKL- 1-3-3-3-5



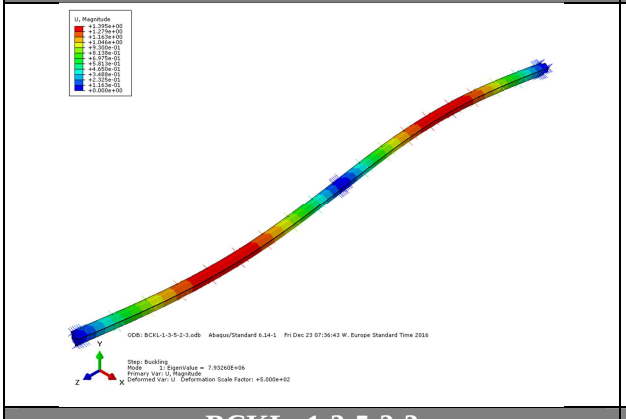
BCKL- 1-3-5-1-3



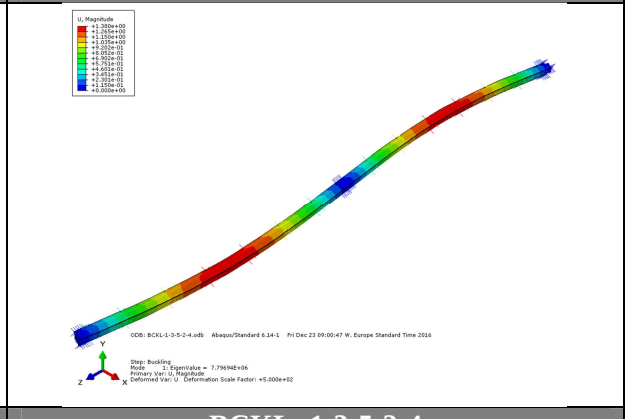
BCKL- 1-3-5-1-4



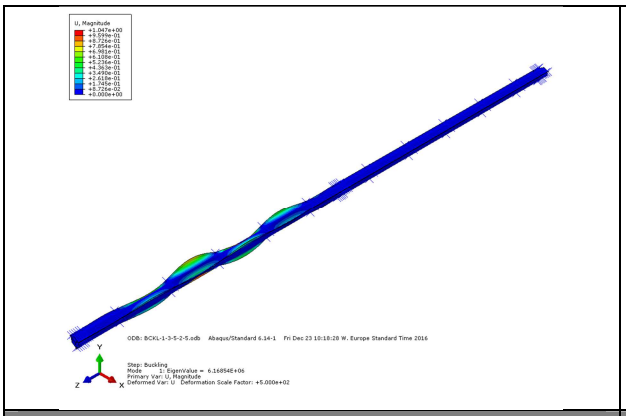
BCKL- 1-3-5-1-5



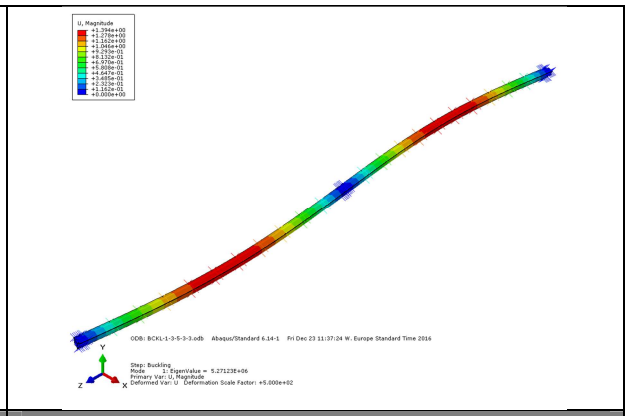
BCKL- 1-3-5-2-3



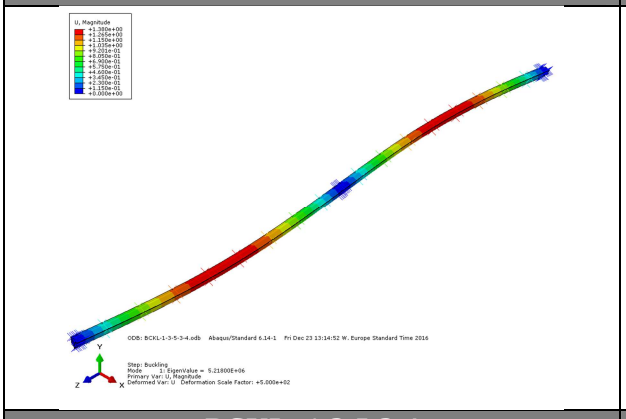
BCKL- 1-3-5-2-4



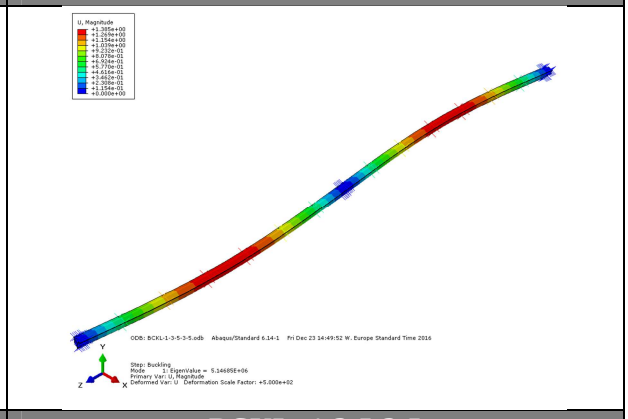
BCKL- 1-3-5-2-5



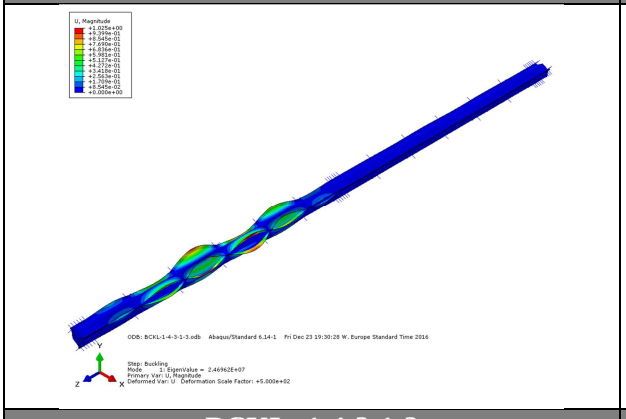
BCKL- 1-3-5-3-3



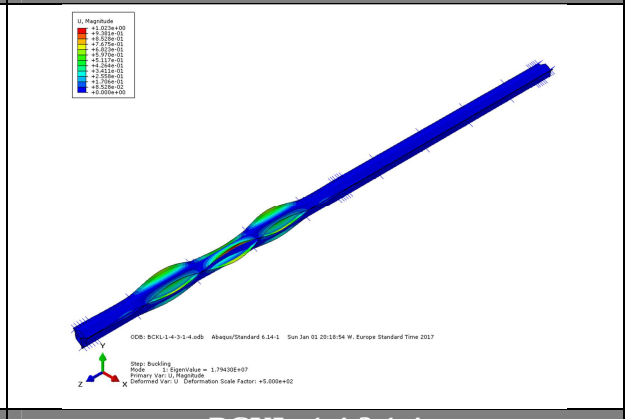
BCKL- 1-3-5-3-4



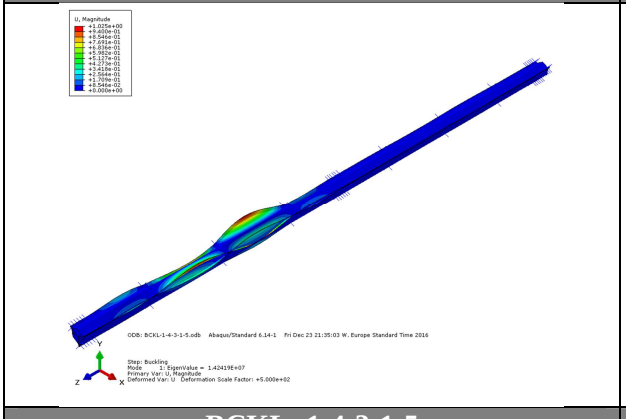
BCKL- 1-3-5-3-5



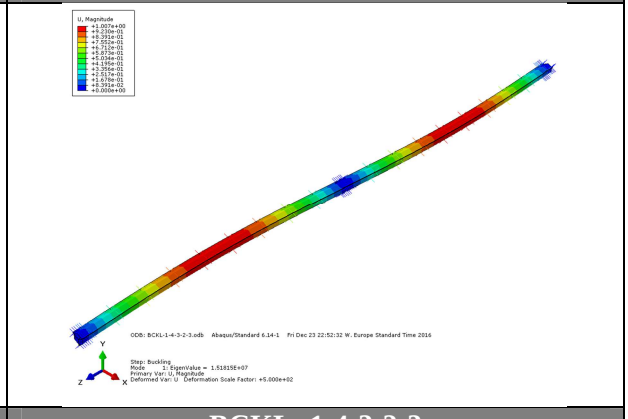
BCKL- 1-4-3-1-3



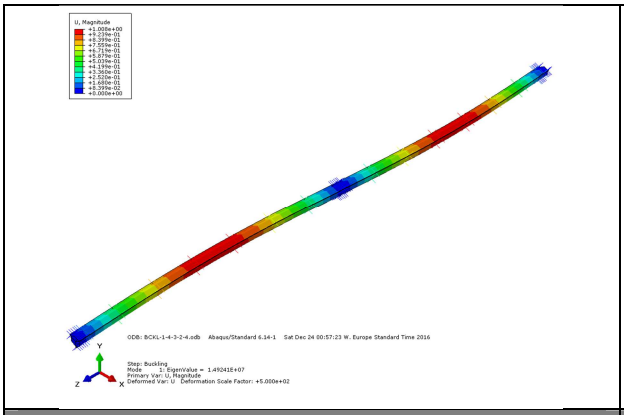
BCKL- 1-4-3-1-4



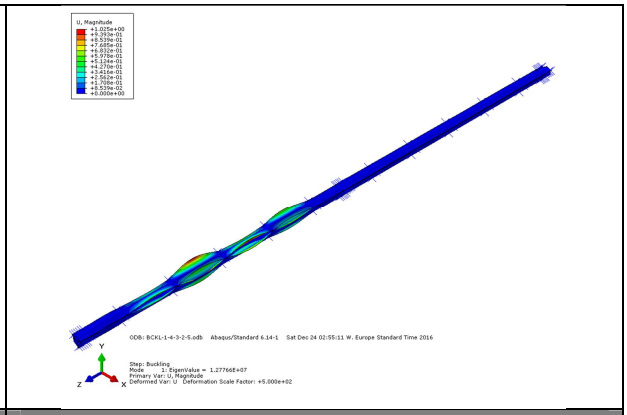
BCKL- 1-4-3-1-5



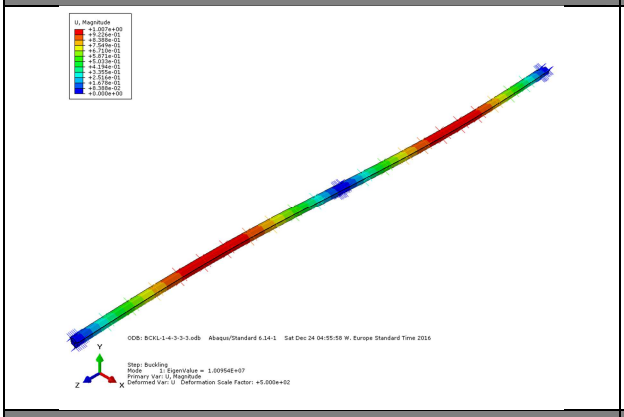
BCKL- 1-4-3-2-3



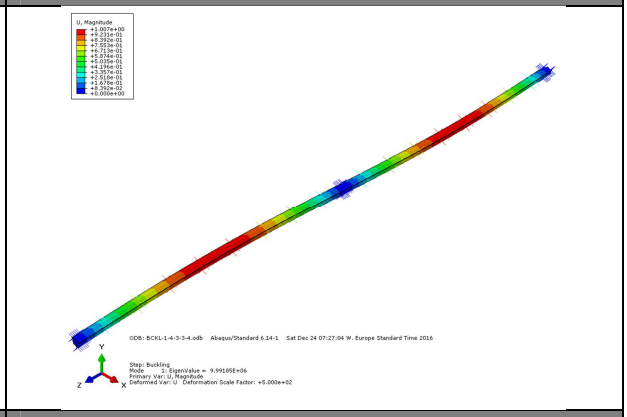
BCKL- 1-4-3-2-4



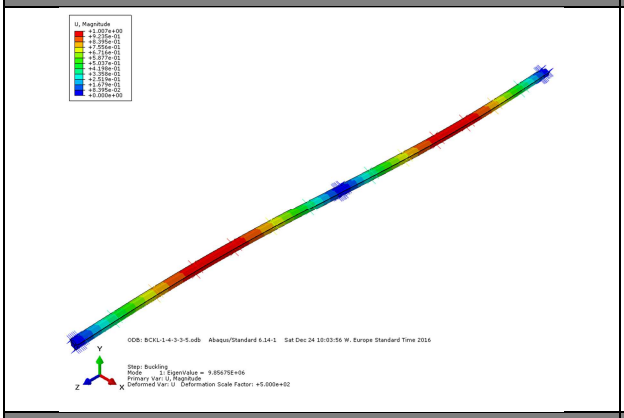
BCKL- 1-4-3-2-5



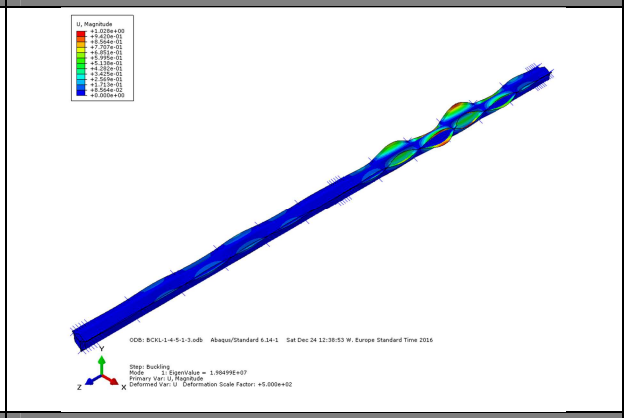
BCKL- 1-4-3-3-3



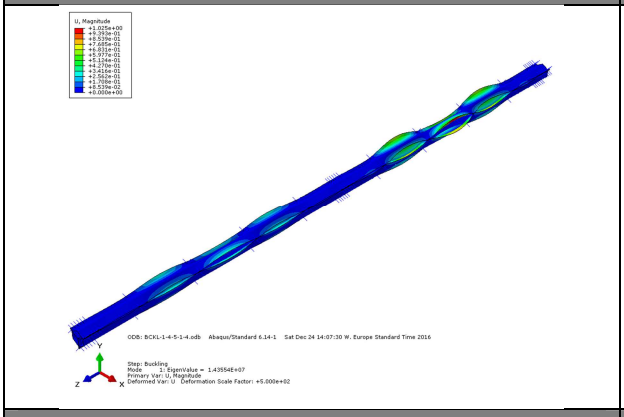
BCKL- 1-4-3-3-4



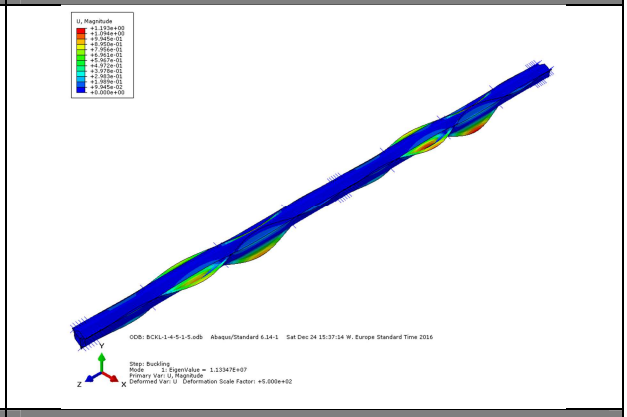
BCKL- 1-4-3-3-5



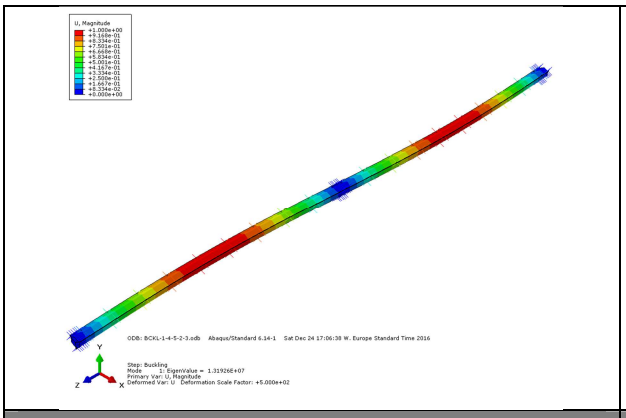
BCKL- 1-4-5-1-3



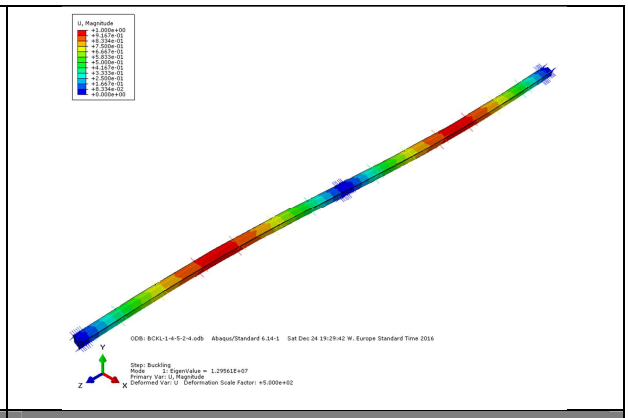
BCKL- 1-4-5-1-4



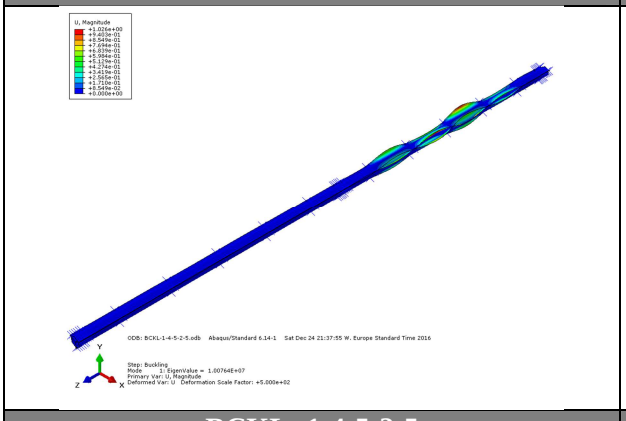
BCKL- 1-4-5-1-5



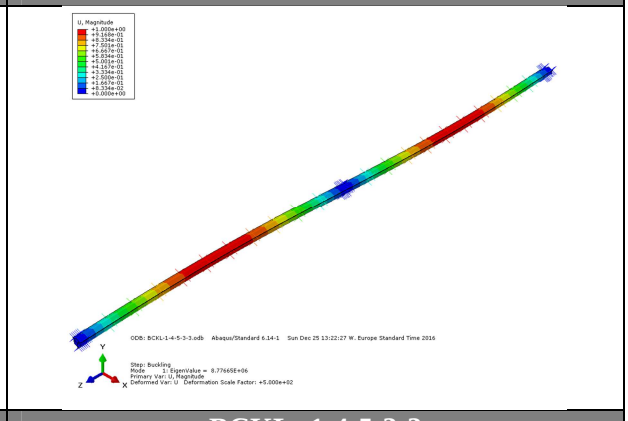
BCKL- 1-4-5-2-3



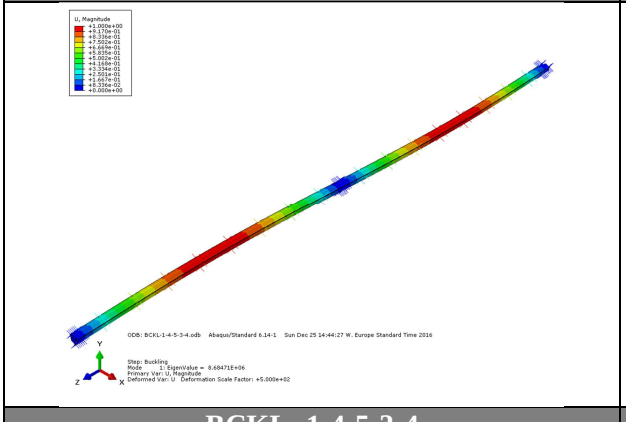
BCKL- 1-4-5-2-4



BCKL- 1-4-5-2-5



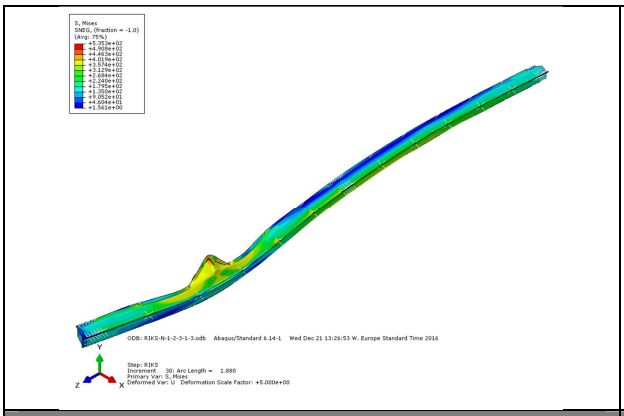
BCKL- 1-4-5-3-3



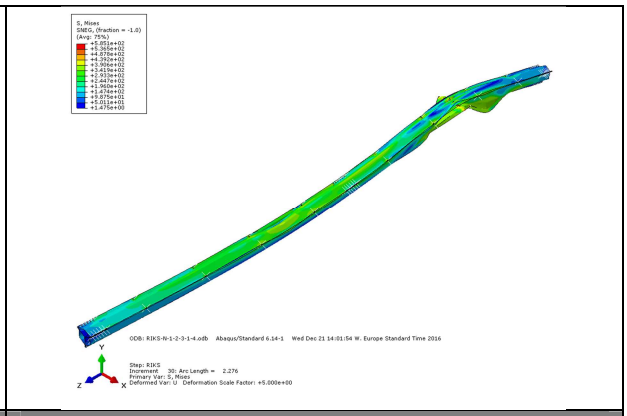
BCKL- 1-4-5-3-4

BCKL- 1-4-5-3-5

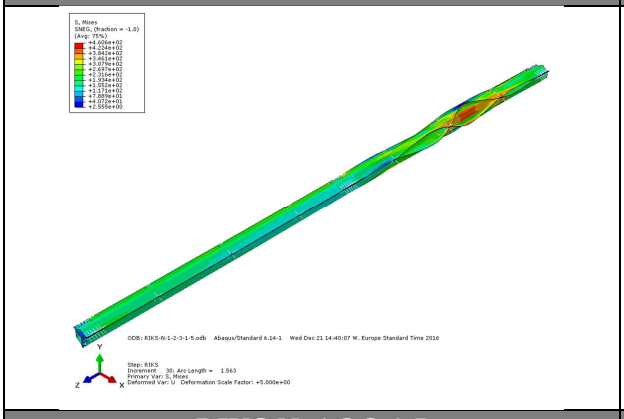
ANNEX C.2. FAILURE MODES FROM RIKS ANALYSIS – RIKS N



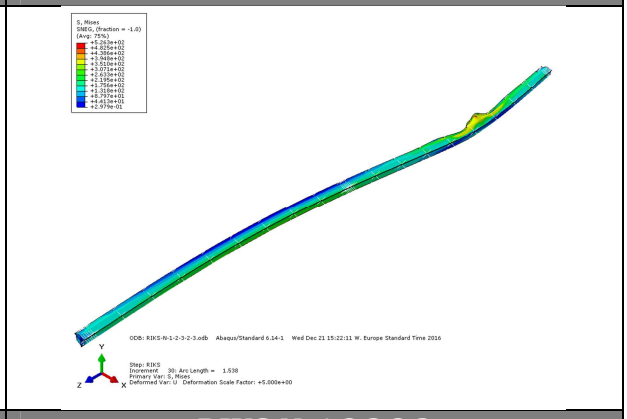
RIKS-N- 1-2-3-1-3



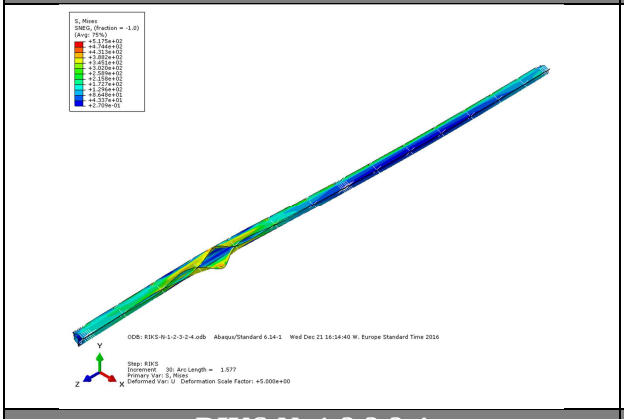
RIKS-N- 1-2-3-1-4



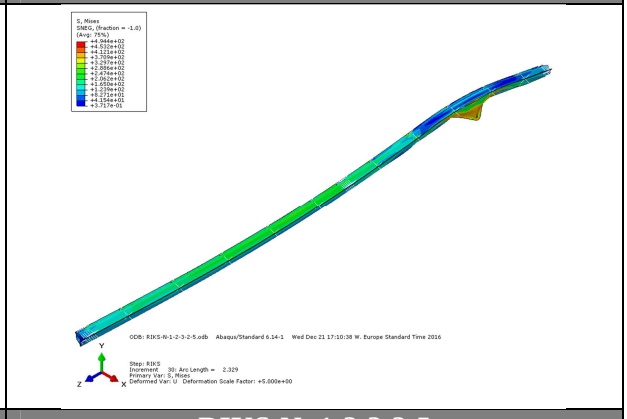
RIKS-N- 1-2-3-1-5



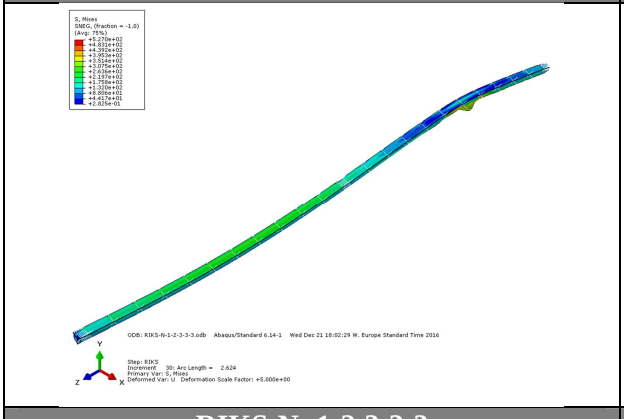
RIKS-N- 1-2-3-2-3



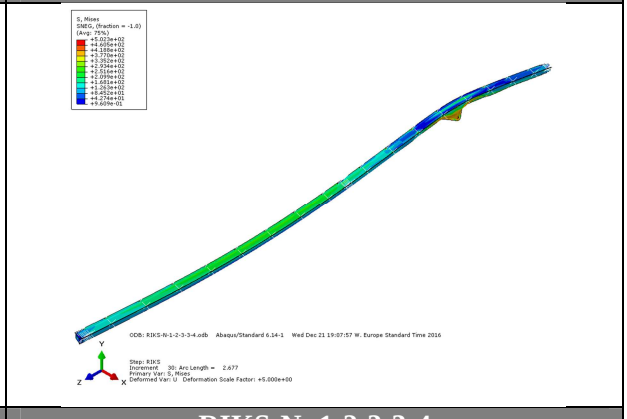
RIKS-N- 1-2-3-2-4



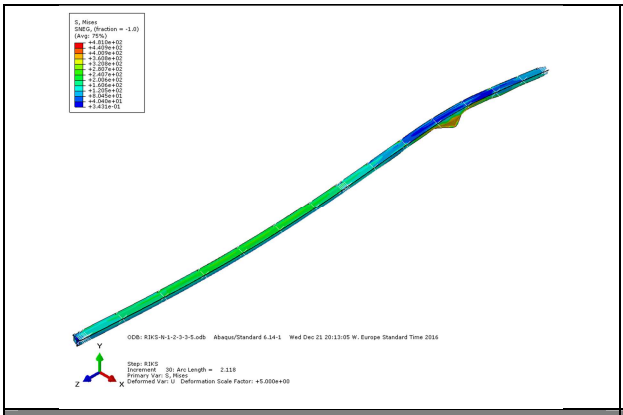
RIKS-N- 1-2-3-2-5



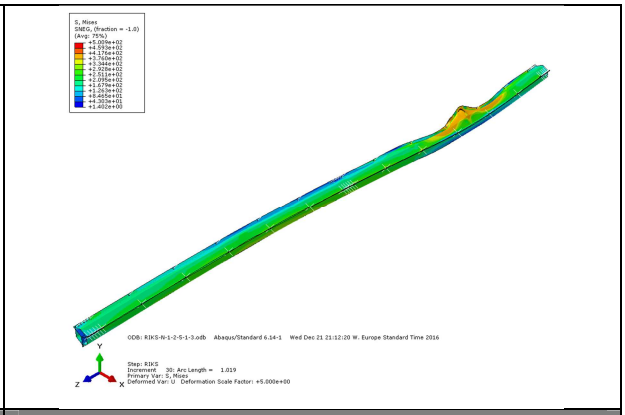
RIKS-N- 1-2-3-3-3



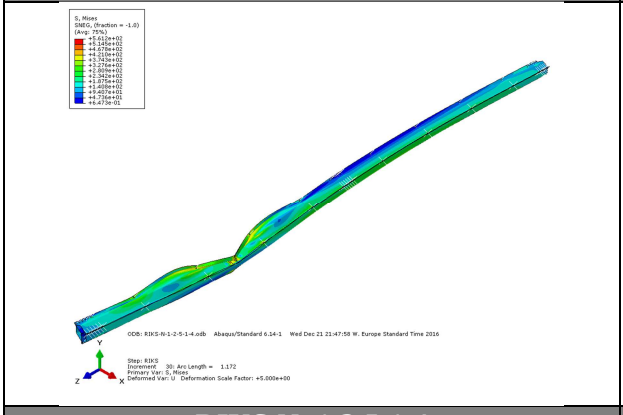
RIKS-N- 1-2-3-3-4



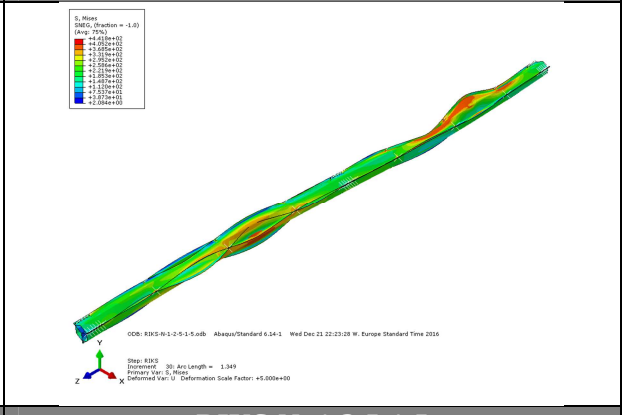
RIKS-N- 1-2-3-3-5



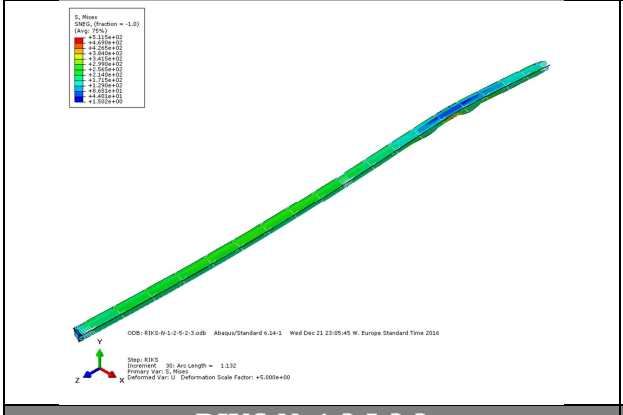
RIKS-N- 1-2-5-1-3



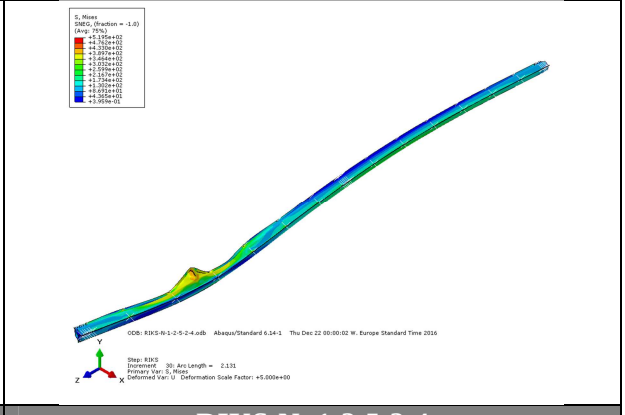
RIKS-N- 1-2-5-1-4



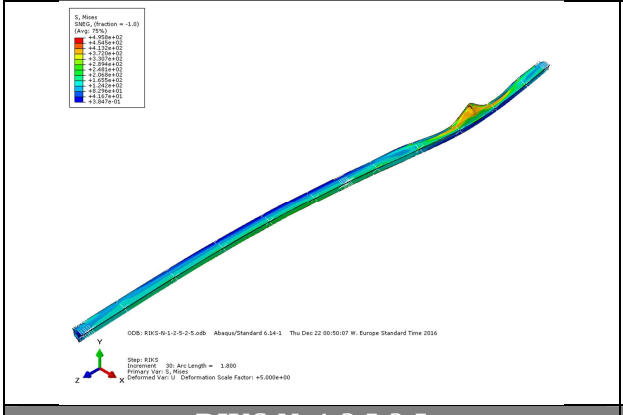
RIKS-N- 1-2-5-1-5



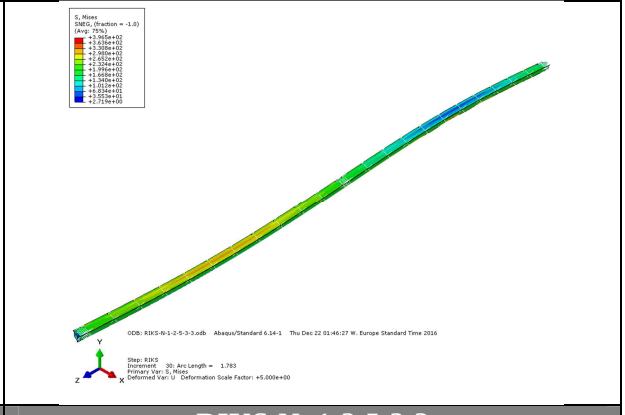
RIKS-N- 1-2-5-2-3



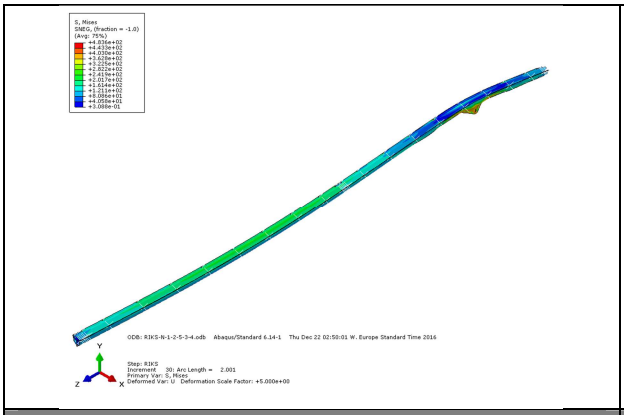
RIKS-N- 1-2-5-2-4



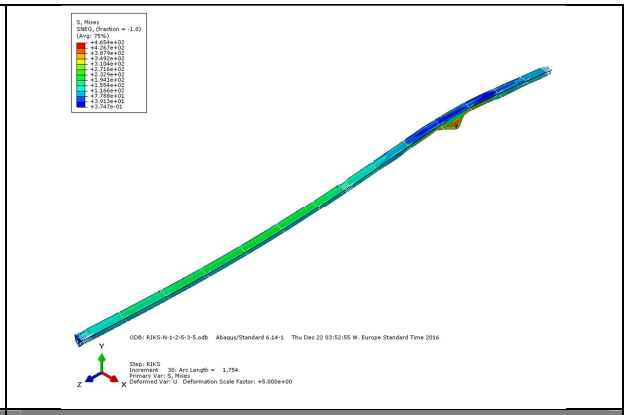
RIKS-N- 1-2-5-2-5



RIKS-N- 1-2-5-3-3

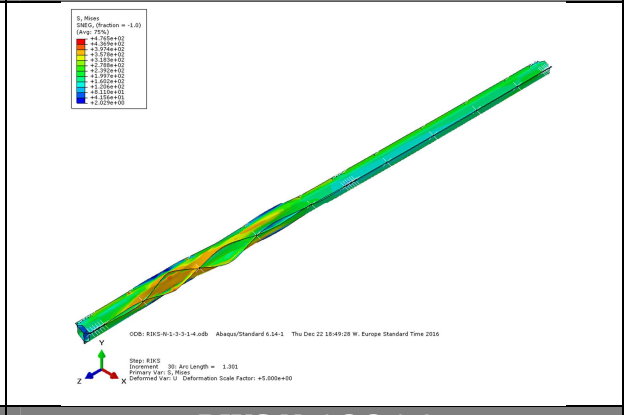


RIKS-N- 1-2-5-3-4

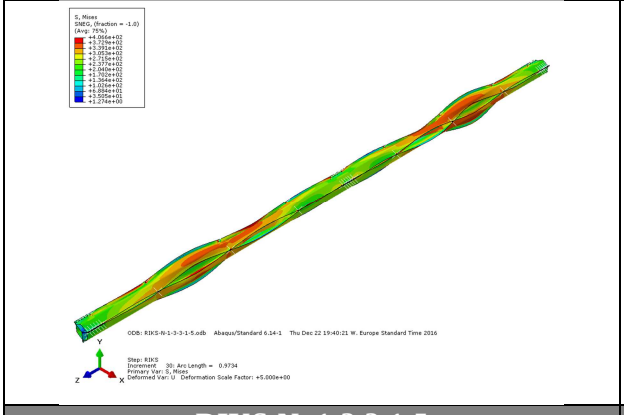


RIKS-N- 1-2-5-3-5

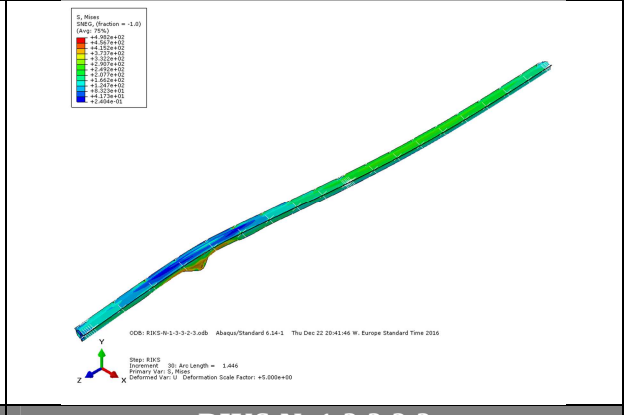
RIKS-N- 1-3-3-1-3



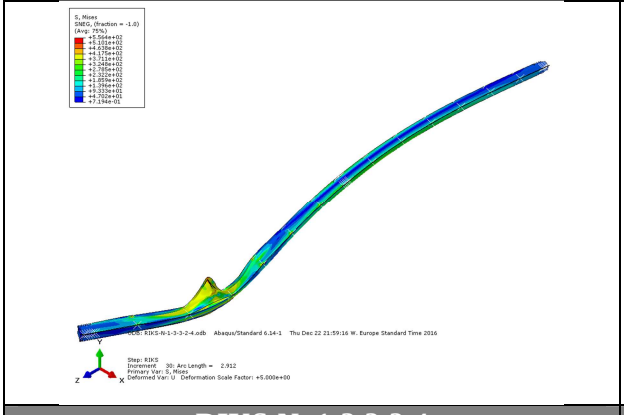
RIKS-N- 1-3-3-1-4



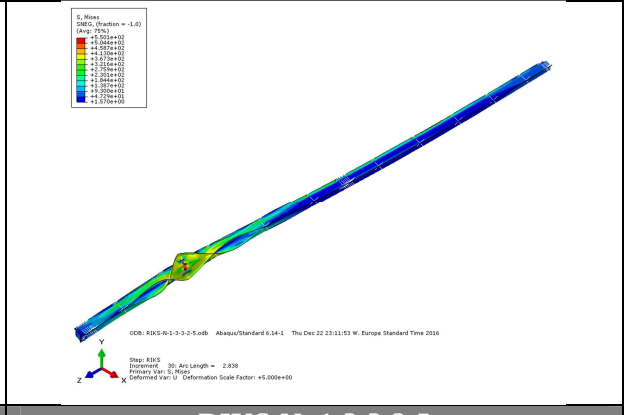
RIKS-N- 1-3-3-1-5



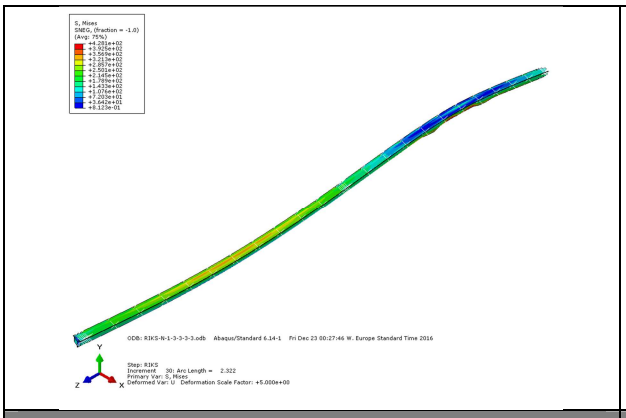
RIKS-N- 1-3-3-2-3



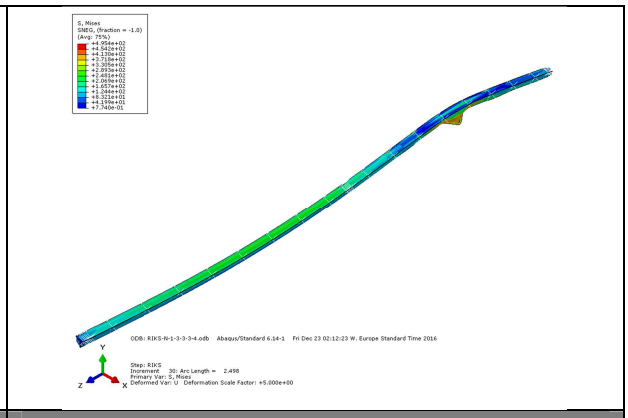
RIKS-N- 1-3-3-2-4



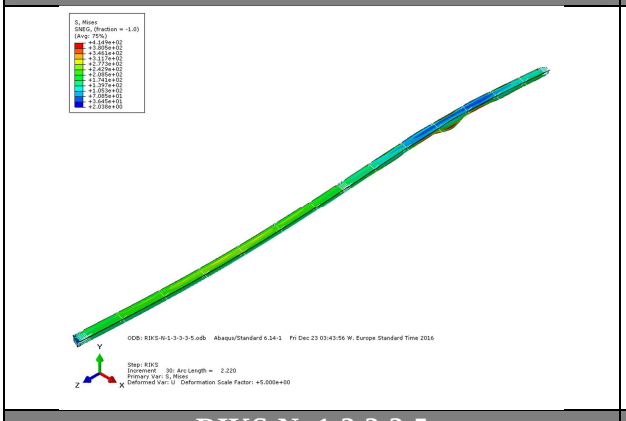
RIKS-N- 1-3-3-2-5



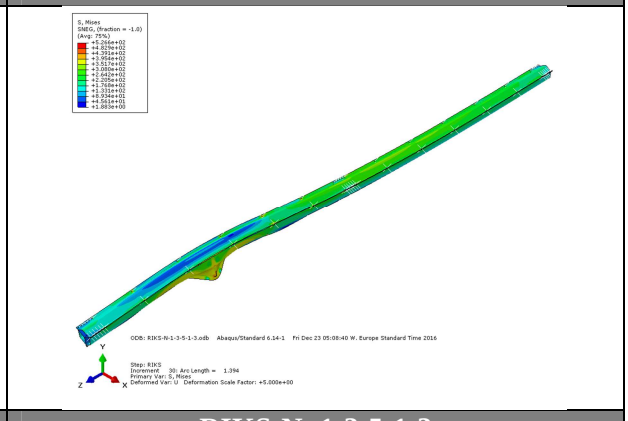
RIKS-N- 1-3-3-3-3



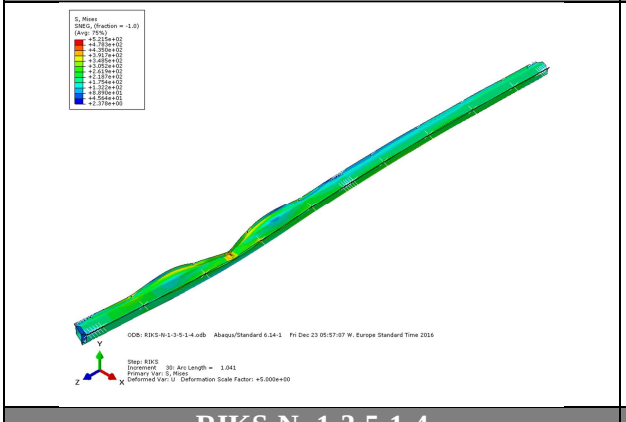
RIKS-N- 1-3-3-3-4



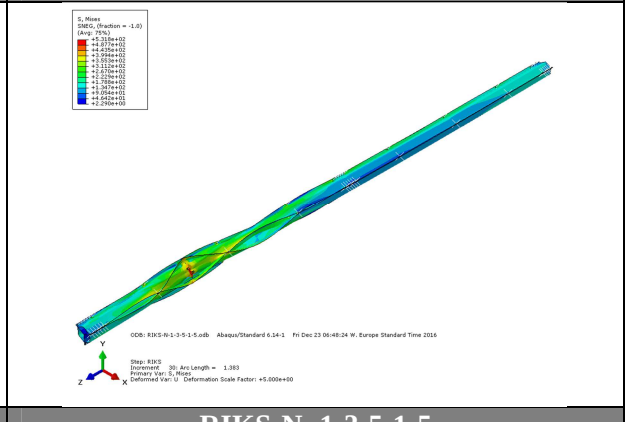
RIKS-N- 1-3-3-3-5



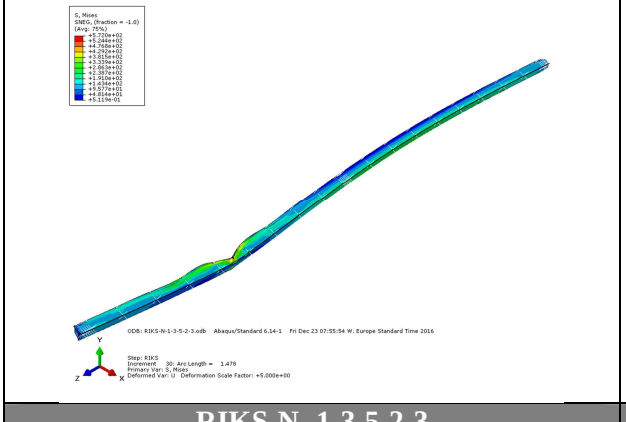
RIKS-N- 1-3-5-1-3



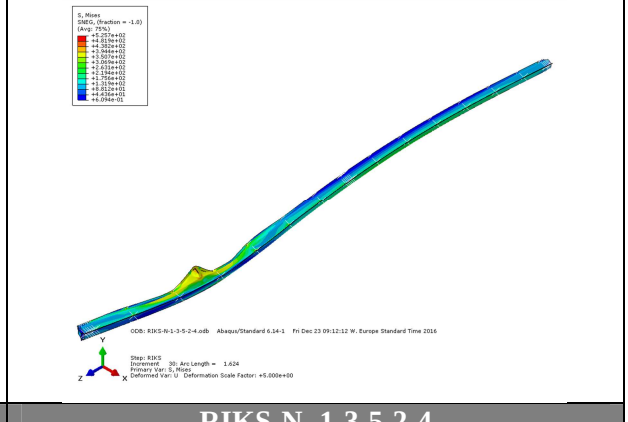
RIKS-N- 1-3-5-1-4



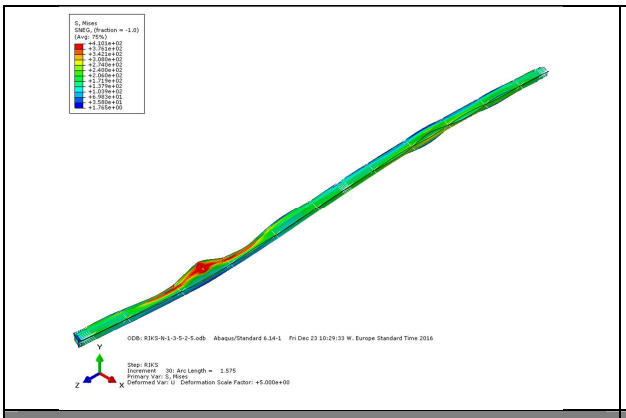
RIKS-N- 1-3-5-1-5



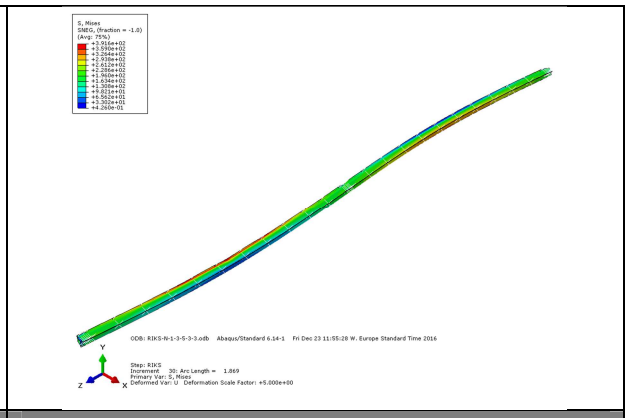
RIKS-N- 1-3-5-2-3



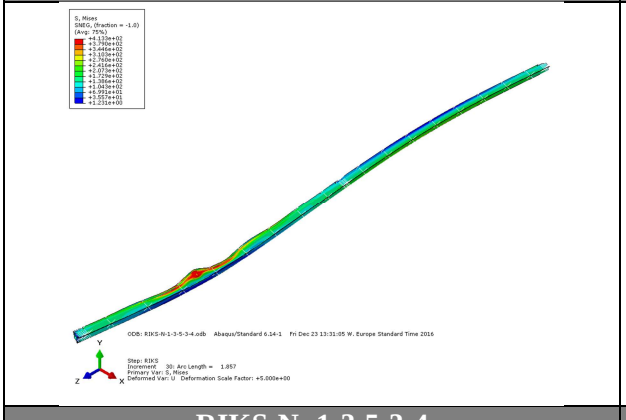
RIKS-N- 1-3-5-2-4



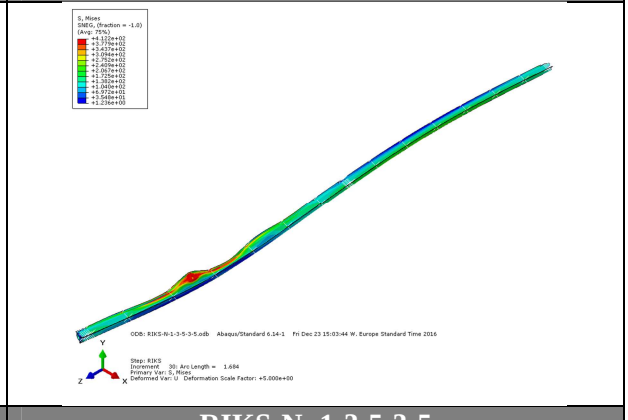
RIKS-N- 1-3-5-2-5



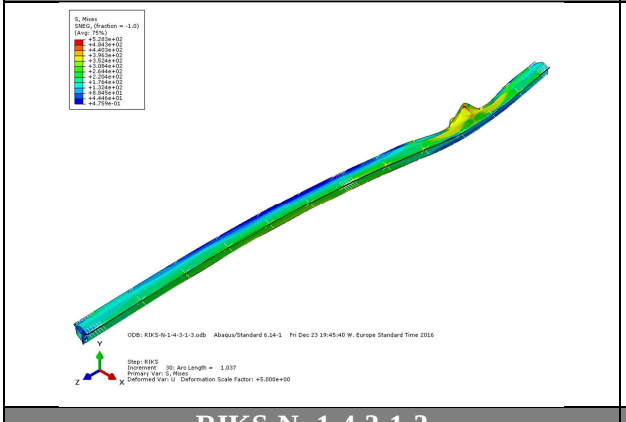
RIKS-N- 1-3-5-3-3



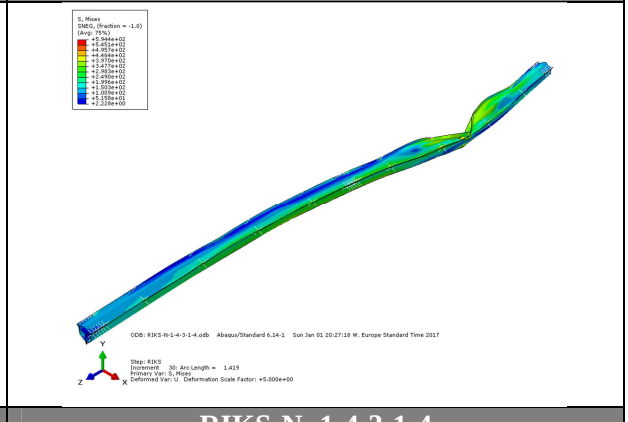
RIKS-N- 1-3-5-3-4



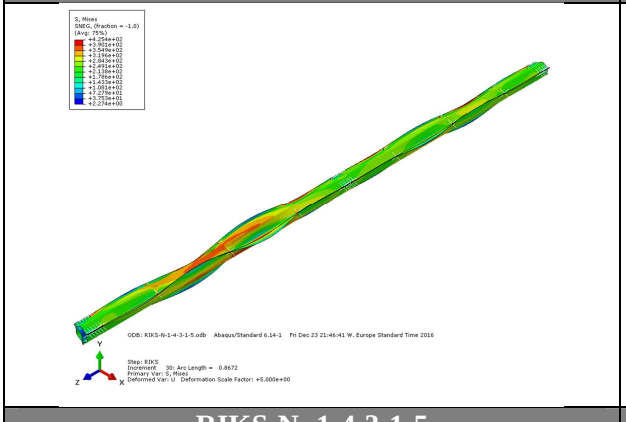
RIKS-N- 1-3-5-3-5



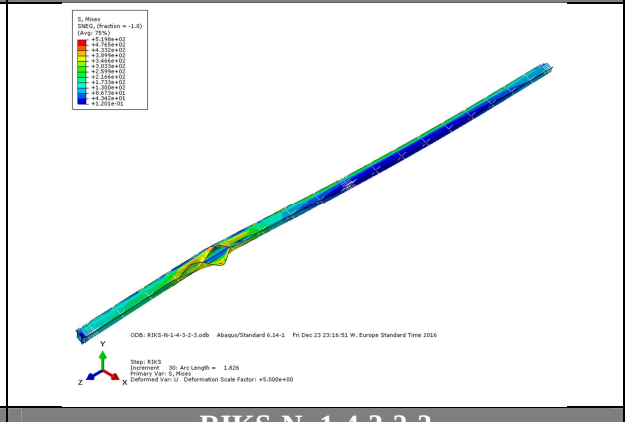
RIKS-N- 1-4-3-1-3



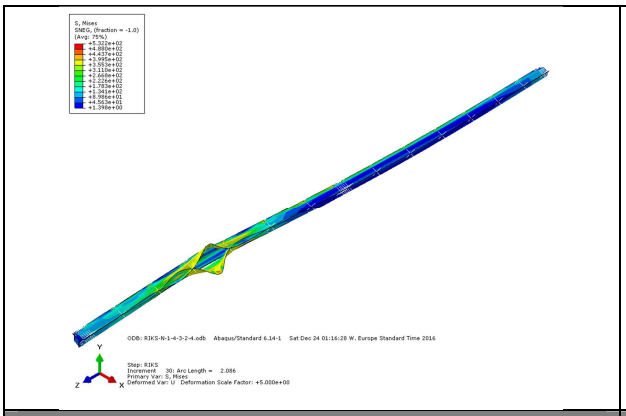
RIKS-N- 1-4-3-1-4



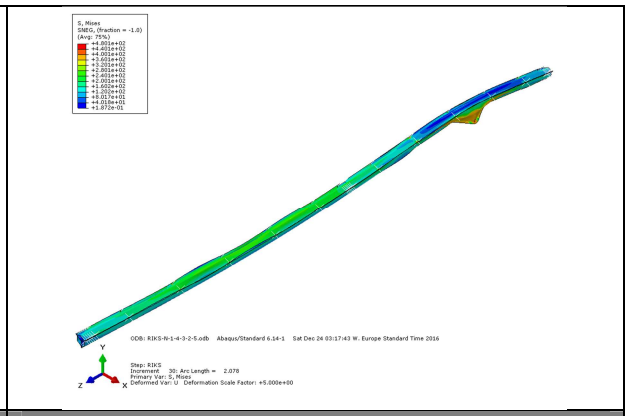
RIKS-N- 1-4-3-1-5



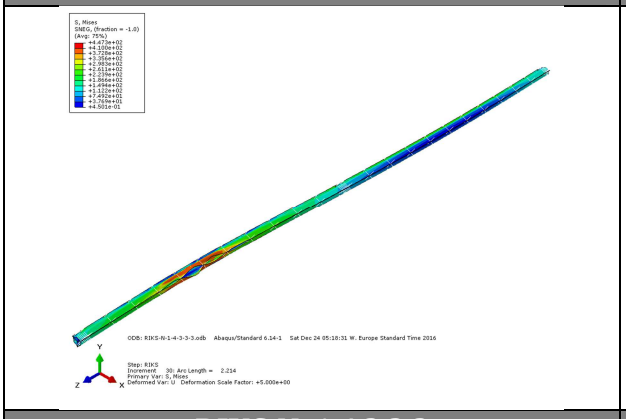
RIKS-N- 1-4-3-2-3



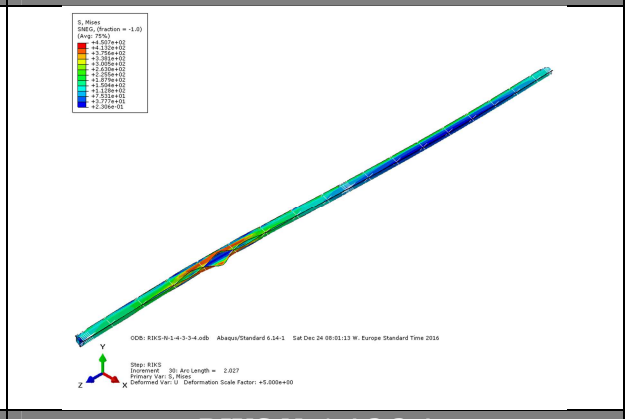
RIKS-N- 1-4-3-2-4



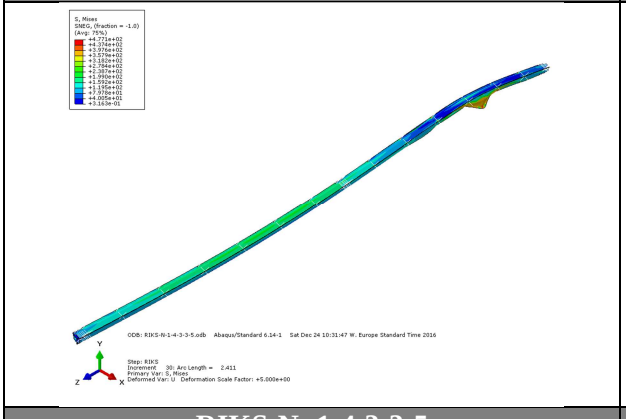
RIKS-N- 1-4-3-2-5



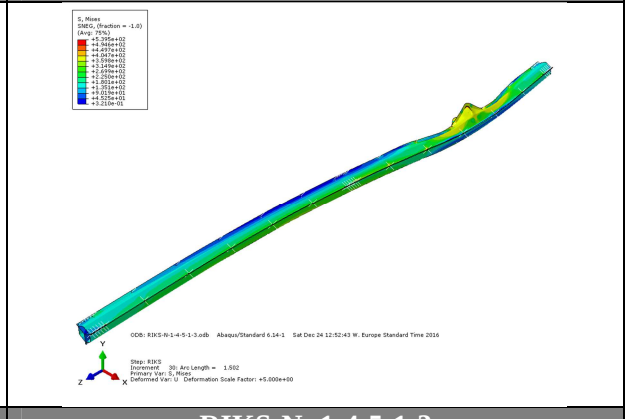
RIKS-N- 1-4-3-3-3



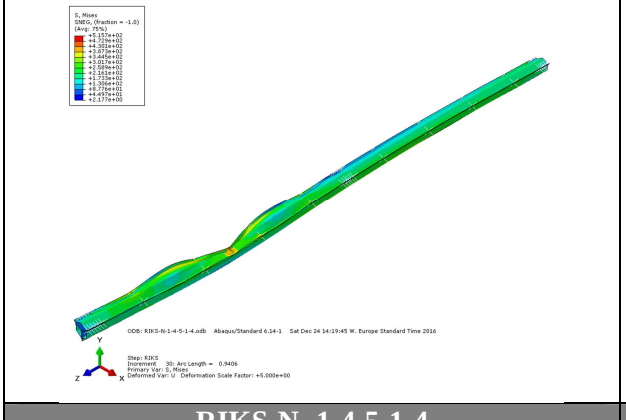
RIKS-N- 1-4-3-3-4



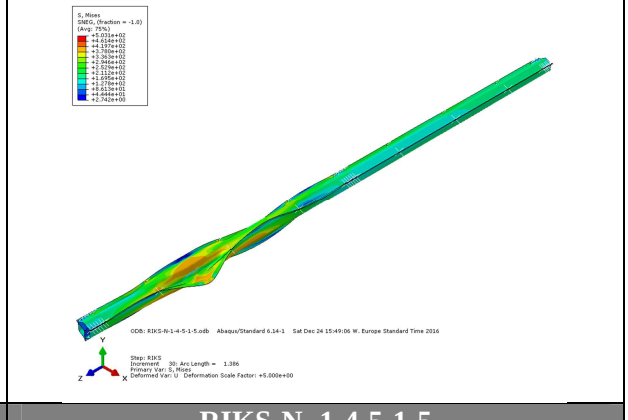
RIKS-N- 1-4-3-3-5



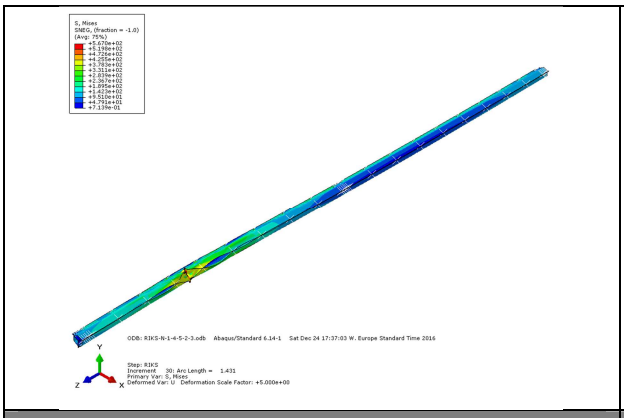
RIKS-N- 1-4-5-1-3



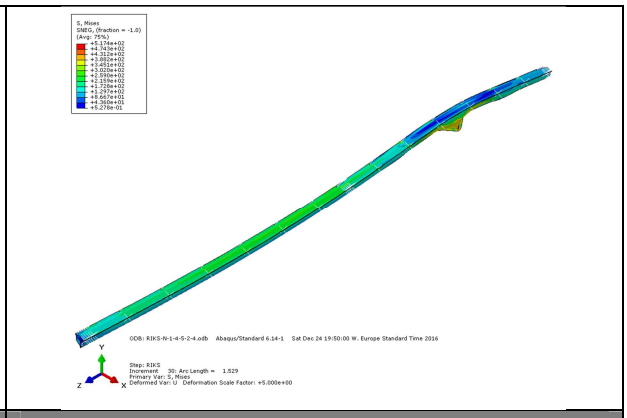
RIKS-N- 1-4-5-1-4



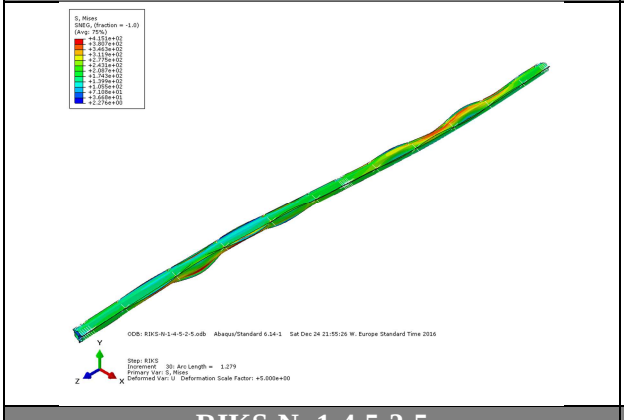
RIKS-N- 1-4-5-1-5



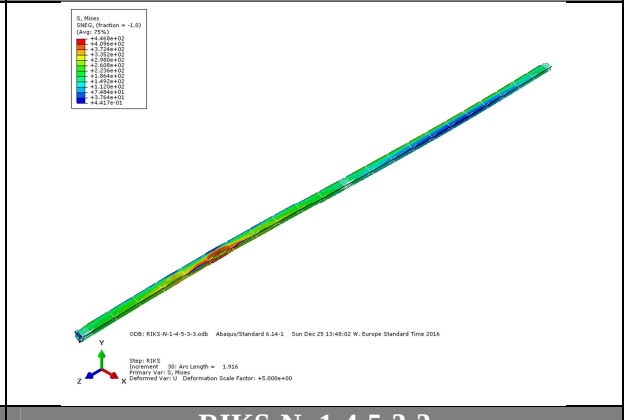
RIKS-N- 1-4-5-2-3



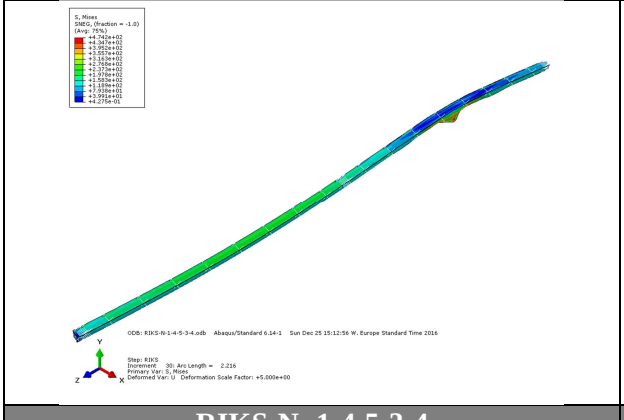
RIKS-N- 1-4-5-2-4



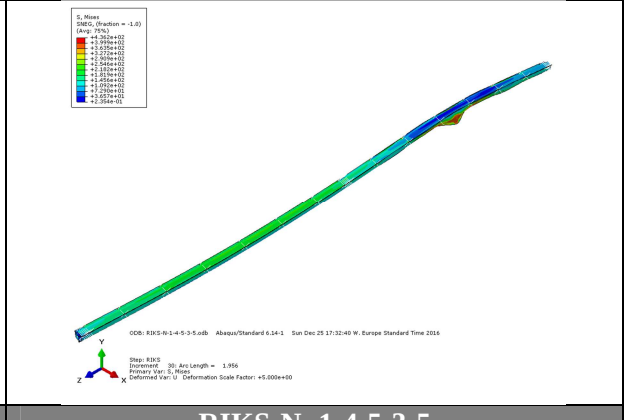
RIKS-N- 1-4-5-2-5



RIKS-N- 1-4-5-3-3



RIKS-N- 1-4-5-3-4



RIKS-N- 1-4-5-3-5

ANNEX D. Results from ODBRead.py for RIKS-N Model

Model ID					d	t	l	d/t	lambda	s/d	Ultimate load	Displ.	Real cs-slend	Area	Class EC1-5	Eff. area local	χ_{dist}	Eff. area distort	Class EC1-6	N_{cRd_local}	N_{cRd_dist}	χ_F	χ_T	χ_{FT}	N_{bRd}	N_{cRd_shell}	f_{ya}	Mass	N_{Rd}/mass
n	d	t	l	b	*[mm]*	*[mm]*	*[mm]*	*[-]*	*[-]*	*[-]*	*[N]*	*[mm]*	*[-]*	*[mm²]*	*[-]*	*[mm²]*	*[-]*	*[mm²]*	*[-]*	*[N]*	*[N]*	*[-]*	*[-]*	*[-]*	*[N]*	*[N]*	*[MPa]*	*[kg]*	*[kN/kg]*
1	2	3	1	3	500	9	18098	55.6	0.65	3	5625706	29.13	83.9	15700	3	15700	0.90	14096	3	5573434	5003923	0.82	1.0	0.82	4070662	5573434	390.8	2230	2.52
1	3	3	1	3	700	12	25339	58.3	0.65	3	10583087	41.10	88.1	29348	3	29348	0.88	25894	3	10418590	9192417	0.82	1.0	0.82	7575642	10418590	389.0	5838	1.81
1	4	3	1	3	900	15	32580	60.0	0.65	3	16670033	51.85	90.6	47204	3	47204	0.87	41277	4	16757418	14653327	0.82	1.0	0.82	12152574	16757418	388.0	12072	1.38
1	2	5	1	3	500	7	18106	71.4	0.65	3	4132538	27.34	107.9	12298	3	12298	0.85	10491	4	4365948	3724270	0.82	1.0	0.82	3111015	4365948	382.6	1748	2.36
1	3	5	1	3	700	10	25347	70.0	0.65	3	8309604	38.52	105.7	24582	3	24582	0.85	20916	4	8726522	7425069	0.82	1.0	0.82	6231490	8726522	383.2	4891	1.70
1	4	5	1	3	900	13	32588	69.2	0.65	3	13828574	49.28	104.6	41073	3	41073	0.85	34926	4	14580769	12398696	0.82	1.0	0.82	10423917	14580769	383.5	10507	1.32
1	2	3	2	3	500	9	27035	55.6	1	3	4865735	37.93	83.9	15700	3	15700	0.88	13879	3	5573434	4927056	0.61	1.0	0.61	2945890	5573434	390.8	3332	1.46
1	3	3	2	3	700	12	37852	58.3	1	3	8974794	52.52	88.1	29348	3	29348	0.87	25461	3	10418590	9038614	0.61	1.0	0.61	5491181	10418590	389.0	8720	1.03
1	4	3	2	3	900	15	48669	60.0	1	3	14330121	66.74	90.6	47204	3	47204	0.86	40540	4	16757418	14391859	0.61	1.0	0.61	8817113	16757418	388.0	18034	0.79
1	2	5	2	3	500	7	27048	71.4	1	3	3665155	36.35	107.9	12298	3	12298	0.84	10282	4	4365948	3650204	0.62	1.0	0.62	2271261	4365948	382.6	2611	1.40
1	3	5	2	3	700	10	37865	70.0	1	3	7105255	49.55	105.7	24582	3	24582	0.83	20319	4	8726522	7213206	0.62	1.0	0.62	4546017	8726522	383.2	7307	0.97
1	4	5	2	3	900	13	48682	69.2	1	3	12124579	65.03	104.6	41073	3	41073	0.83	34221	4	14580769	12148284	0.62	1.0	0.62	7601410	14580769	383.5	15696	0.77
1	2	3	3	3	500	9	33419	55.6	1.25	3	3335747	34.09	83.9	15700	3	15700	0.54	8507	3	5573434	3019951	0.47	1.0	0.47	2255339	5573434	390.8	4119	0.81
1	3	3	3	3	700	12	46790	58.3	1.25	3	6192939	47.06	88.1	29348	3	29348	0.54	15856	3	10418590	5628825	0.47	1.0	0.47	4207730	10418590	389.0	10780	0.57
1	4	3	3	3	900	15	60161	60.0	1.25	3	9958434	60.53	90.6	47204	3	47204	0.54	25420	4	16757418	9023969	0.47	1.0	0.47	6759878	16757418	388.0	22293	0.45
1	2	5	3	3	500	7	33435	71.4	1.25	3	2606744	33.78	107.9	12298	3	12298	0.54	6673	4	4365948	2368985	0.48	1.0	0.48	1747482	4365948	382.6	3228	0.81
1	3	5	3	3	700	10	46806	70.0	1.25	3	5177700	47.83	105.7	24582	3	24582	0.54	13268	4	8726522	4710080	0.48	1.0	0.48	3496152	8726522	383.2	9032	0.57
1	4	5	3	3	900	13	60177	69.2	1.25	3	8656932	61.02	104.6	41073	3	41073	0.54	22102	4	14580769	7846074	0.48	1.0	0.48	5844575	14580769	383.5	19402	0.45
1	2	3	1	4	500	9	18098	55.6	0.65	4	4982469	26.03	83.9	15700	3	15700	0.81	12683	3	5573434	4502428	0.82	1.0	0.82	4070662	5573434	390.8	2230	2.23
1	3	3	1	4	700	12	25339	58.3	0.65	4	9053748	35.59	88.1	29348	3	29348	0.78	23037	3	10418590	8178229	0.82	1.0	0.82	7575642	10418590	389.0	5838	1.55
1	4	3	1	4	900	15	32580	60.0	0.65	4	14571561	45.18	90.6	47204	3	47204	0.77	36408	4	16757418	12924900	0.82	1.0	0.82	12152574	16757418	388.0	12072	1.21
1	2	5	1	4	500	7	18106	71.4	0.65	4	3576703	24.21	107.9	12298	3	12298	0.75	9235	4	4365948	3278378	0.82	1.0	0.82	3111015	4365948	382.6	1748	2.05
1	3	5	1	4	700	10	25347	70.0	0.65	4	7358898	34.13	105.7	24582	3	24582	0.75	18318	4	8726522	6502901	0.82	1.0	0.82	6231490	8726522	383.2	4891	1.50
1	4	5	1	4	900	13	32588	69.2	0.65	4	11788016	43.61	104.6	41073	3	41073	0.74	30449	4	14580769	10809407	0.82	1.0	0.82	10423917	14580769	383.5	10507	1.12
1	2	3	2	4	500	9	27035	55.6	1	4	4554955	35.44	83.9	15700	3	15700	0.78	12304	3	5573434	4367961	0.61	1.0	0.61	2945890	5573434	390.8	3332	1.37
1	3	3	2	4	700	12	37852	58.3	1	4	8128143	47.34	88.1	29348	3	29348	0.76	22234	3	10418590	7893145	0.61	1.0	0.61	5491181	10418590	389.0	8720	0.93
1	4	3	2	4	900	15	48669	60.0	1	4	13158688	61.34	90.6	47204	3	47204	0.74	35005	4	16757418	12426866	0.61	1.0	0.61	8817113	16757418	388.0	18034	0.73

Model ID					d	t	l	d/t	lambda	s/d	Ultimate load	Displ.	Real cs-slend	Area	Class EC1-5	Eff. area local	χ_{dist}	Eff. area distort	Class EC1-6	N_{cRd_local}	N_{cRd_dist}	χ_F	χ_T	χ_{FT}	N_{bRd}	N_{cRd_shell}	f_{ya}	Mass	N_{Rd}/mass
n	d	t	l	b	*[mm]*	*[mm]*	*[mm]*	*[-]*	*[-]*	*[-]*	*[N]*	*[mm]*	*[-]*	*[mm²]*	*[-]*	*[mm²]*	*[-]*	*[mm²]*	*[-]*	*[N]*	*[N]*	*[-]*	*[-]*	*[-]*	*[N]*	*[N]*	*[MPa]*	*[kg]*	*[kN/kg]*
1	2	5	2	4	500	7	27048	71.4	1	4	3241900	32.20	107.9	12298	3	12298	0.72	8854	4	4365948	3143084	0.62	1.0	0.62	2271261	4365948	382.6	2611	1.24
1	3	5	2	4	700	10	37865	70.0	1	4	6373937	44.55	105.7	24582	3	24582	0.71	17524	4	8726522	6220994	0.62	1.0	0.62	4546017	8726522	383.2	7307	0.87
1	4	5	2	4	900	13	48682	69.2	1	4	10701573	57.72	104.6	41073	3	41073	0.71	29076	4	14580769	10322023	0.62	1.0	0.62	7601410	14580769	383.5	15696	0.68
1	2	3	3	4	500	9	33419	55.6	1.25	4	3269501	33.10	83.9	15700	3	15700	0.78	12204	3	5573434	4332464	0.47	1.0	0.47	2255339	5573434	390.8	4119	0.79
1	3	3	3	4	700	12	46790	58.3	1.25	4	6020515	45.09	88.1	29348	3	29348	0.75	22020	3	10418590	7817013	0.47	1.0	0.47	4207730	10418590	389.0	10780	0.56
1	4	3	3	4	900	15	60161	60.0	1.25	4	9777493	58.85	90.6	47204	3	47204	0.73	34629	4	16757418	12293238	0.47	1.0	0.47	6759878	16757418	388.0	22293	0.44
1	2	5	3	4	500	7	33435	71.4	1.25	4	2496823	31.14	107.9	12298	3	12298	0.71	8752	4	4365948	3107017	0.48	1.0	0.48	1747482	4365948	382.6	3228	0.77
1	3	5	3	4	700	10	46806	70.0	1.25	4	4985356	44.02	105.7	24582	3	24582	0.70	17314	4	8726522	6146646	0.48	1.0	0.48	3496152	8726522	383.2	9032	0.55
1	4	5	3	4	900	13	60177	69.2	1.25	4	8400347	57.10	104.6	41073	3	41073	0.70	28703	4	14580769	10189730	0.48	1.0	0.48	5844575	14580769	383.5	19402	0.43
1	2	3	1	5	500	9	18098	55.6	0.65	5	4531789	24.75	83.9	15700	3	15700	0.73	11501	3	5573434	4082844	0.82	1.0	0.82	4070662	5573434	390.8	2230	2.03
1	3	3	1	5	700	12	25339	58.3	0.65	5	8140806	33.86	88.1	29348	3	29348	0.70	20655	3	10418590	7332567	0.82	1.0	0.82	7575642	10418590	389.0	5838	1.39
1	4	3	1	5	900	15	32580	60.0	0.65	5	12630036	41.79	90.6	47204	3	47204	0.69	32370	4	16757418	11491298	0.82	1.0	0.82	12152574	16757418	388.0	12072	1.05
1	2	5	1	5	500	7	18106	71.4	0.65	5	3161856	23.20	107.9	12298	3	12298	0.67	8204	4	4365948	2912516	0.82	1.0	0.82	3111015	4365948	382.6	1748	1.81
1	3	5	1	5	700	10	25347	70.0	0.65	5	6417546	32.86	105.7	24582	3	24582	0.66	16159	4	8726522	5736437	0.82	1.0	0.82	6231490	8726522	383.2	4891	1.31
1	4	5	1	5	900	13	32588	69.2	0.65	5	10664361	42.53	104.6	41073	3	41073	0.71	29339	4	14580769	10415366	0.82	1.0	0.82	10423917	14580769	383.5	10507	1.01
1	2	3	2	5	500	9	27035	55.6	1	5	4126212	32.17	83.9	15700	3	15700	0.70	10959	3	5573434	3890319	0.61	1.0	0.61	2945890	5573434	390.8	3332	1.24
1	3	3	2	5	700	12	37852	58.3	1	5	7400332	43.45	88.1	29348	3	29348	0.66	19486	3	10418590	6917544	0.61	1.0	0.61	5491181	10418590	389.0	8720	0.85
1	4	3	2	5	900	15	48669	60.0	1	5	11412761	53.88	90.6	47204	3	47204	0.64	30305	4	16757418	10758139	0.61	1.0	0.61	8817113	16757418	388.0	18034	0.63
1	2	5	2	5	500	7	27048	71.4	1	5	2840900	28.59	107.9	12298	3	12298	0.62	7663	4	4365948	2720258	0.62	1.0	0.62	2271261	4365948	382.6	2611	1.09
1	3	5	2	5	700	10	37865	70.0	1	5	5661079	40.01	105.7	24582	3	24582	0.61	14996	4	8726522	5323710	0.62	1.0	0.62	4546017	8726522	383.2	7307	0.77
1	4	5	2	5	900	13	48682	69.2	1	5	9216875	50.32	104.6	41073	3	41073	0.60	24655	4	14580769	8752650	0.62	1.0	0.62	7601410	14580769	383.5	15696	0.59
1	2	3	3	5	500	9	33419	55.6	1.25	5	3176238	31.49	83.9	15700	3	15700	0.69	10807	3	5573434	3836554	0.47	1.0	0.47	2255339	5573434	390.8	4119	0.77
1	3	3	3	5	700	12	46790	58.3	1.25	5	5901916	43.72	88.1	29348	3	29348	0.65	19151	3	10418590	6798664	0.47	1.0	0.47	4207730	10418590	389.0	10780	0.55
1	4	3	3	5	900	15	60161	60.0	1.25	5	9320637	54.80	90.6	47204	3	47204	0.63	29705	4	16757418	10545433	0.47	1.0	0.47	6759878	16757418	388.0	22293	0.42
1	2	5	3	5	500	7	33435	71.4	1.25	5	2405325	30.03	107.9	12298	3	12298	0.61	7506	4	4365948	2664540	0.48	1.0	0.48	1747482	4365948	382.6	3228	0.75
1	3	5	3	5	700	10	46806	70.0	1.25	5	4682954	40.90	105.7	24582	3	24582	0.60	14666	4	8726522	5206291	0.48	1.0	0.48	3496152	8726522	383.2	9032	0.52
1	4	5	3	5	900	13	60177	69.2	1.25	5	7913537	53.21	104.6	41073	3	41073	0.59	24052	4	14580769	8538333	0.48	1.0	0.48	5844575	14580769	383.5	19402	0.41

ANNEX E. List of Available Profiles

Model					**Diameter**	**Thickness**	**Length**	**Area of section**	**Mass per meter**	**Moment of inertia about y- and z-axis**		**Radius of gyration**	**Torsional constant**	**Warping constant**
i	j	k	l	b	D	T	L	A	M	I_{xx}	I_{yy}	i_y	I_t	I_w
[-]	*[-]*	*[-]*	*[-]*	*[-]*	*[mm]*	*[mm]*	*[mm]*	*[mm^2] x10^2*	*[kg/m]*	*[mm^4] x10^6*	*[mm^4] x10^6*	*[mm]*	*[mm^4] x10^6*	*[mm^6] x10^{12}*
1	2	3	1	3	500	9	18098	157.00	123.24	438.34	438.34	170.92	138.36	71.72
1	3	3	1	3	700	12	25339	293.48	230.38	1606.29	1606.29	240.01	419.90	515.53
1	4	3	1	3	900	15	32580	472.04	370.55	4271.20	4271.20	309.15	951.58	2267.05
1	2	5	1	3	500	7	18106	122.98	96.54	343.73	343.73	173.83	151.87	56.42
1	3	5	1	3	700	10	25347	245.82	192.97	1346.41	1346.41	243.00	459.43	433.13
1	4	5	1	3	900	13	32588	410.73	322.42	3718.53	3718.53	312.18	1033.09	1977.27
1	2	3	2	3	500	9	27035	157.00	123.24	438.34	438.34	170.92	138.36	71.72
1	3	3	2	3	700	12	37852	293.48	230.38	1606.29	1606.29	240.01	419.90	515.53
1	4	3	2	3	900	15	48669	472.04	370.55	4271.20	4271.20	309.15	951.58	2267.05
1	2	5	2	3	500	7	27048	122.98	96.54	343.73	343.73	173.83	151.87	56.42
1	3	5	2	3	700	10	37865	245.82	192.97	1346.41	1346.41	243.00	459.43	433.13
1	4	5	2	3	900	13	48682	410.73	322.42	3718.53	3718.53	312.18	1033.09	1977.27
1	2	3	3	3	500	9	33419	157.00	123.24	438.34	438.34	170.92	138.36	71.72
1	3	3	3	3	700	12	46790	293.48	230.38	1606.29	1606.29	240.01	419.90	515.53
1	4	3	3	3	900	15	60161	472.04	370.55	4271.20	4271.20	309.15	951.58	2267.05
1	2	5	3	3	500	7	33435	122.98	96.54	343.73	343.73	173.83	151.87	56.42
1	3	5	3	3	700	10	46806	245.82	192.97	1346.41	1346.41	243.00	459.43	433.13
1	4	5	3	3	900	13	60177	410.73	322.42	3718.53	3718.53	312.18	1033.09	1977.27