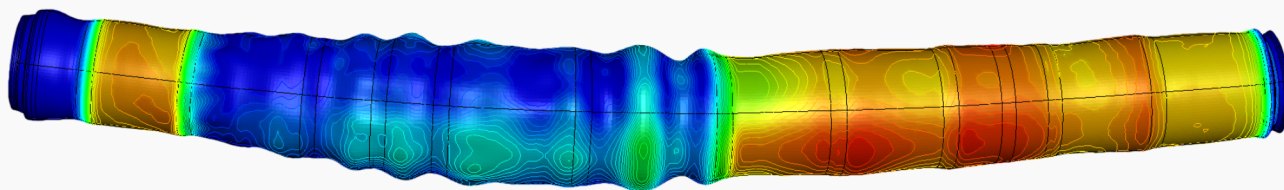


Upgrades of the Target Dump Externals for the Large Hadron Collider

Downstream flange and upgraded weld configuration



Asbjørn Lassen Lund

7th semester, fall 2022

Aarhus University, Denmark
School of engineering
Faculty of Mechanical engineering

21. december 2022



Copyright © Aarhus University 2022

This bachelor thesis was typeset with L^AT_EX using Overleaf. The title page illustration was taken from a simulation in LS-Dyna[®]. Figures have been created using Microsoft Excel, Plotly or Matplotlib in Python and flowcharts using draw.io. Figures from simulations are outputs from LS-Dyna[®] and Ansys Mechanical[®].

Number of normal pages: 37.6



Abstract

The Large Hadron Collider provides high-energy particles for fundamental research at The European Organization for Nuclear Research. The comprehensive complex is continuously upgraded to allow increased particle beam energy and is expected to reach 680 MJ towards High Luminosity in 2027. The Target Dumps Externals are essential security equipment capable of absorbing the energy of the beams. In this thesis, a new downstream flange design and vessel weld configuration are evaluated for the construction of the Target Dump External spares to ensure a safe design capable of withstanding the loads upon beam absorption. Using explicit finite element analysis, upgrades are optimized for increased performance, highlighting design challenges and proposals for improvements.



Asbjørn Lassen Lund



Preface

This is a 7th semester bachelor thesis by Asbjørn Lassen Lund in mechanical engineering at Aarhus University, Denmark. The thesis and related work were compiled from August 2022 to December 2022.

Reading Guide

It is expected that the reader has general knowledge in the area of mechanical engineering, including the application of Finite Element Analysis.

The bachelor thesis follows the guidelines established by the ISO/IEC 80000 standard. SI units are used unless otherwise specified. Period (.) is used as decimal separator and space as thousand separator. Citations are made according to the IEEE citation method. Figures without citations are created by the author for the project. A list of acronyms used in the thesis and their meanings is found before the table of contents. Figures, tables, and equations are inserted and enumerated according to chapter and order of insertion.

The report is structured as follows: initially, an introduction to the engineering problem is presented, highlighting the challenges and design criteria for the project. The following chapters introduce CERN and its accelerator complex and describe the function and location of the Large Hadron Collider Target Dump Externals. They also provide an overview of the evolution of its design. This is followed by a brief presentation of explicit Finite Element Analysis methods, which in the following chapters are the main tool to assess the upgraded aspects of the Target Dump Externals. Finally, the outcome of the project is presented and ideas for further assessment are given.

Acknowledgements

Major thanks to the Target Dump External team of the Sources, Target and Collimators group at CERN and my supervisor Nicola Solieri for engineering guidance and insightful discussions on different aspects of the work presented in this bachelor thesis. Big thanks to my professors Martin Heide Jørgensen and Martin Nielsen from Aarhus University for supervising the project and giving advice throughout the semester.



Acronyms

CERN - The European Organization for Nuclear Research

LHC - Large Hadron Collider

PSB - Proton Synchrotron Booster

PS - Proton Synchrotron

SPS - Super Proton Synchrotron

TDE - Target Dump External

FLUKA - Fluktuierende Kaskade

FEA - Finite Element Analysis

FEM - Finite Element Method

CPU - Central Processing Unit

RAM - Random Access Memory

CFL - Courant Friedrichs Lewy

GUI - Graphical User Interface

APDL - Ansys Parametric Design Language

DR - Dynamic Relaxation

GMAW - Gas Metal Arc Welding

MPP - Massively Parallel Processing

SMP - Symmetric Multi Processing

FFT - Fast Fourier Transformation

LDV - Laser Doppler velocimetry



Contents

1	Introduction	1
2	Problem	2
3	The European Organization for Nuclear Research	3
3.1	The Science Program	3
3.2	The Accelerator Complex	3
4	The Target Dump External for the Large Hadron Collider	6
4.1	Beam Dump Procedure	7
4.2	Composition of the Target Dump External	8
4.3	Response Upon Beam Impact	11
4.4	Modifications for the Run 3 Spares	12
5	Finite Element Method	15
5.1	Implicit vs. Explicit Algorithms	15
5.2	Explicit Algorithm Stability Condition	17
6	Simulation of Dynamic Response Upon Beam Impact	19
6.1	Energy Density Map Using FLUKA	20
6.2	Simulation of Shrink Fitting	21
6.3	Simulation of Beam Impact	25
6.4	Conclusion	29
7	Downstream Flange	30
7.1	Simulation of the Downstream Flange	32
7.2	Damping of Simulation Data	37
7.3	Multi Axial Fatigue	40
7.4	Conclusion	44
8	Circumferential Welds	45
8.1	Eurocode 3 Static Verification	46
8.2	Post Processing of Simulation Results	47

8.3	Welding Performance Qualification Record (WPQR) according to ISO15613 of Uranus 45	51
8.4	Tensile Tests Simulation of Weld	52
8.5	3D Submodel Simulation of Weld	56
8.6	2D Axisymmetric Submodel Simulation of Weld	57
8.7	Conclusion	60
9	Discussion	62
10	Conclusion	64
A	Appendix	66
A.1	Equivalent Beam Energy Cases	67
A.2	Geometry Preparation and Simplifications of the TDE for Simulation	68
A.3	Meshing Guidelines	72
A.4	Shrink Fitting Calculations (ANSI/ASME B4.1)	73
A.5	Geometry Preparation and Simplifications of The Downstream Flange for Simulation .	74
A.6	Eurocode 3 Tables for Static Verification	76
A.7	Python Script for Eurocode 3 Static Verification	77
A.8	APDL Script for Implementation of Boundary Displacements In 2D Axisymmetric Simulation of Weld	89
A.9	Python Script for Estimation of Damping Ratio Using LDV Data	90
	Bibliography	92

Introduction

The Large Hadron Collider at the European Organization for Nuclear Research provides proton beams for high-energy physics experiments. Using superconducting electromagnets to accelerate and steer the particle beams in a 27 km circular trajectory, the particles reach 99.99 % of the speed of light. At this velocity, the beams contain 539 MJ of energy equivalent to the kinetic energy stored in a 68-ton airplane flying at 450 km/h.

A comprehensive safety system protects the accelerator in case of equipment failure. The Target Dump Externals (TDEs) are dedicated beam absorbers capable of safely absorbing the particle beams. These devices consist of 8.5 m Ø720 mm diameter duplex stainless steel vessels containing graphitic material cores of different densities, in a Nitrogen inner atmosphere. They are located 100 m underground in purpose-built caverns at the end of 600 m extraction lines tangential to the Large Hadron Collider ring.

Upon beam impact, the TDEs are subject to extreme thermomechanical loads that challenge the structural integrity of the components. Due to the increased stored beam energy of the beams. Two operational spares are currently being built for these objects and their design must be validated to ensure safe operation during their lifetime.

During Run 2 (2015 to 2018), the downstream flange of the TDEs was found to leak Nitrogen, potentially causing oxidation of the graphite core. Furthermore, the weld configuration of the vessel had not previously been assessed and is expected due to the increased beam energy in Run 3 to be a weak aspect in the design of these devices. For the construction of the Run 3 spares beginning in early 2023, an updated design must be validated to ensure the capability of withstanding the loads.

Problem

As a consequence of increased beam energy in the Large Hadron Collider, the design and upgrades of the Target Dump Externals for the manufacturing of the Run 3 spares beginning in early 2023 need to be validated to ensure safe operation during their lifetime. This thesis is limited to an assessment of the modified downstream flange design and the circumferential welds along the length of the Target Dump Externals.

The modified flange design and the circumferential welds along the vessel must:

- Withstand beam dumps with energy up to 539 *MJ*.
- Withstand 4 years of operation with 400 full-intensity beam dumps per year.
- Be compliant with current manufacturing methods.
- Integrate easily into the current design.

The European Organization for Nuclear Research

The European Organization for Nuclear Research (CERN), is a European research facility located at the Franco-Swiss border north of the city of Geneva. The objective of CERN is to substantiate the understanding of the composition and physical processes governing the laws of matter, of which the universe is composed. CERN was established in 1954 by 12 member states, including Denmark, at the end of the Second World War [15].

3.1 The Science Program

Today, more than 17 500 people with more than 110 nationalities work together at CERN to push the limits of knowledge. The main scientific focus is particle physics, but the physics program has evolved to include much broader aspects, ranging from nuclear to high-energy physics, from studies of antimatter to the possible effects of cosmic rays on clouds. Common to all aspects included in the program is the motivation to gain a more comprehensive understanding of the fundamental bricks upon which our universe is built [36].

The main purpose of CERN is to provide particle accelerators and infrastructure to enable physicists to perform high-energy physics experiments. The CERN complex has grown over the years and currently operates seven accelerators and two decelerators that can deliver particle beams at the requested intensity and energy level for different experiments. In addition, large data storage and supercomputer facilities are available at CERN to store the many petabytes of experimental data accumulated annually and to perform heavy computing applications such as calculations and simulations [18].

3.2 The Accelerator Complex

The largest accelerator in the CERN accelerator complex is the Large Hadron Collider (LHC). The LHC is located 100 m underground at the Franco-Swiss border and is with its 27 km circumference the world's largest scientific instrument. The LHC is built from resonators and high-field superconducting electromagnets operating at ultra-low vacuum to accelerate and steer the particle beams in a circular trajectory along the circumference of the LHC ring in counter-rotating directions. At four points along the LHC ring, the beams cross-pass creating collisions in the experiments [18]. The LHC is filled

with protons according to a filling scheme that highlights the composition of the bunch train pattern, where protons are divided into bunches forming the beam. An overview of the beam parameters over different accelerator runs can be found in table 3.1 [21].

	Run 1 2009-2013	Run 2 2015-2018	Run 3 2022-2024	HL-LHC 2027-
Beam dumps / year	220 days · 2 = 440			
Energy / proton [TeV]	4.0	6.5	6.8	7.0
Bunch intensity (protons / bunch)	$1.7 \cdot 10^{11}$	$1.2 \cdot 10^{11}$	$1.8 \cdot 10^{11}$	$2.2 \cdot 10^{11}$
Number of bunches / beam	1 380	2 556	2 748	2 760
Stored energy [MJ]	150	320	539	680

Table 3.1: Overview of the beam parameters over different LHC runs. Notice the increase in energy stored over runs [9].

By accelerating and balancing the beams in an electromagnetic field, the beams reach 99.99% of the speed of light. This results in more than 11 000 turns and 1 billion collisions in the LHC per second [28]. Scientists and engineers are seeking ever-increased luminosity, as it results in higher collision rates at the experiments, with a consequently higher physics output. Derived from table 3.1 the energy of the beam is expected to increase more than 450 % from 150 MJ in Run 1 in 2009 to 680 MJ towards High Luminosity scheduled to begin in 2027.

To produce protons, hydrogen gas is injected into a metal cylinder. By applying an electrical field, the hydrogen is broken down into protons. The ion source known as duoplasmatron yields approximately 70% protons [23]. As shown in table 3.1, 2748 bunches of $1.8 \cdot 10^{11}$ protons are present in the LHC during nominal Run 3 operation. In 2016 a total of $1.34 \cdot 10^{20}$ protons were accelerated in the accelerator complex, roughly corresponding to the number of protons in a grain of sand [7].

The LHC relies on a chain of smaller accelerators in the complex that supplies the LHC with beams of the correct injection energy. A map of the CERN accelerator complex is seen in figure 3.1. The particle acceleration begins in the Linear accelerator 4 (Linac4) which accelerates the beams to 160 MeV to prepare them to enter the Proton Synchrotron Booster (PSB). From the PSB, through the Proton Synchrotron (PS), and onto the Super Proton Synchrotron (SPS), the beams are accelerated from 120 MeV to 450 GeV before injection into the two beamlines of the LHC. In the LHC the beams are accelerated to 6.5 TeV before colliding inside four detectors, ALICE, ATLAS, CMS, and LHCb [18].

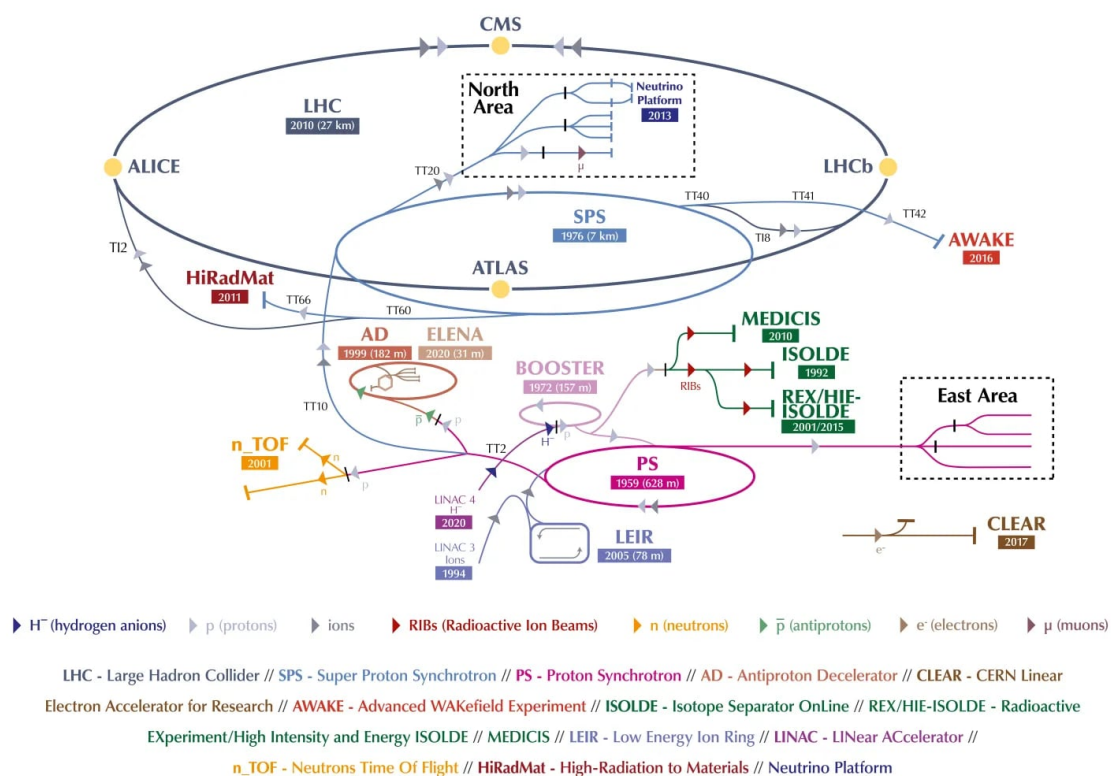


Figure 3.1: The CERN Accelerator Complex [18].

The Target Dump Externals for the Large Hadron Collider

The Target Dump Externals (TDEs) are essential parts of the LHC safety system. They are designed, operated, and maintained by the Targets, Collimators and Dumps (TCD) section in the Sources, Targets and Interactions (STI) group. The TDEs are located at the end of two 600 m long extraction lines located at Point 6 [21]. The location of the TDEs in the CERN Accelerator Complex can be seen in figure 4.1. The TDEs weigh 6.2 tons each and consist of an 8.5 m long and $\varnothing 720$ mm diameter duplex stainless steel vessel containing a core of graphitic materials of different densities [20]. In figure 4.2 one of the TDEs is shown installed in the cavern.

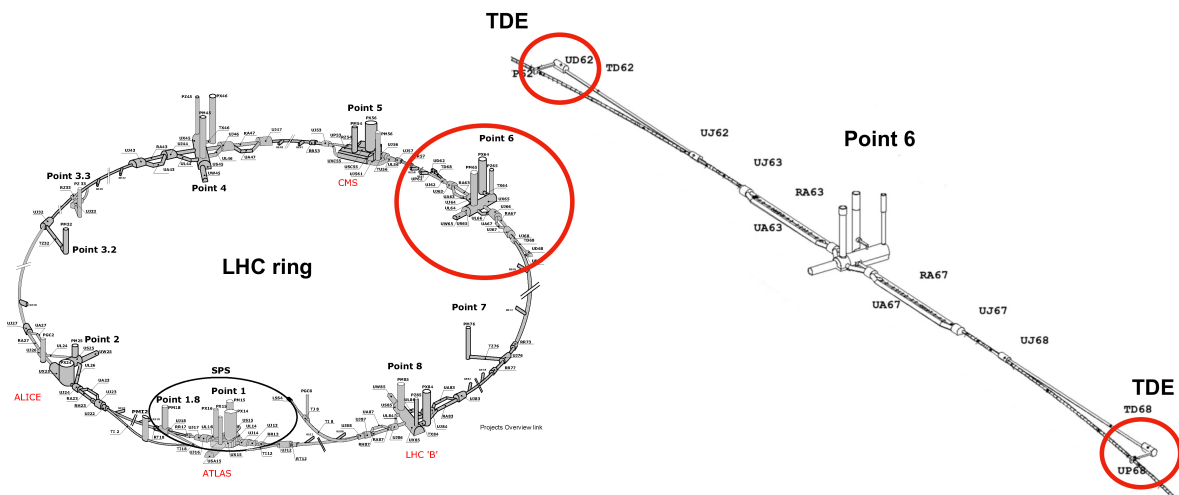


Figure 4.1: The Large Hadron Collider ring. Point 6 enlarged to see the TDEs [14].

The purpose of the TDEs is to safely absorb the energy stored in the LHC in the event of equipment failure in the accelerator or due to a low concentration of protons in the beams, preventing the desired number of collisions at the experiments. The TDEs are the largest beam-interacting devices at CERN. They are subject to high thermomechanical loads upon beam impact due to the high stored energy of the beam and the relatively short absorption length of the dump [21].



(a) Installation of Target Dump External in the cavern 100 m under-ground. (b) Target Dump External installed inside concrete shielding.

Figure 4.2: Installation of the Target Dump External. The beam dump is protected by 750 tons of concrete blocks. The beam travels in free air for a few meters before impacting the upstream window in figure b [25].

Due to the TDEs' role as safety elements for the accelerator, it is critical that they are engineered to withstand the loads during their lifetime. In the event of a failure of the TDEs, the LHC will be unable to run until an exchange of the dumps is carried out, which would have a strong input on the physics output of the machine.

4.1 Beam Dump Procedure

In the event of a beam dump, the beam is extracted from the LHC and into the extraction lines by precise electromagnetic kickers. The beam bunches are injected into the LHC in a pattern that leaves a 3-microsecond gap (the so-called *abortion gap*) between the first and last bunch. This corresponds to the time required for the horizontally deflecting extraction kickers to activate and thus extract the beams from the accelerator [9]. At the beginning of the 600 m long extraction lines, the concentrated beam is diluted into a sweep by two orthogonally deflecting dilution kickers to reduce the peak energy density, ultimately reducing the local loads induced on the TDEs [21]. A plot of the energy deposition density produced by this sweep is shown in figure 4.3.

Up to 539 MJ is absorbed in the beam dump during the $86\text{ }\mu\text{s}$ long impact [20]. This energy level is equivalent to the energy required to heat and melt 875 kg copper or the kinetic energy stored in a 68-ton Airbus A320-100 that flies at 450 km/h(A.1). At impact, the beam protons start to interact with the matter of the TDE core. The probability of the beam protons hitting a nucleus of matter in the cores is proportional to the distance the protons have travelled in the TDE. Upon a collision between the protons and nucleus in the core material, a particle shower of secondary particles is generated, further colliding and interacting with the materials of the core [10]. As a consequence of the interactions, significant heat is generated causing the thermal expansion of the materials. FLUKA simulations, which are Monte Carlo probability-based particle energy deposition simulations [10], have shown that 82% of the beam energy is deposited in the beam dump and the remaining 18% in the 750-ton concrete shielding surrounding the TDEs [21]. The FLUKA Monte Carlo Code is further described in section 6.1.

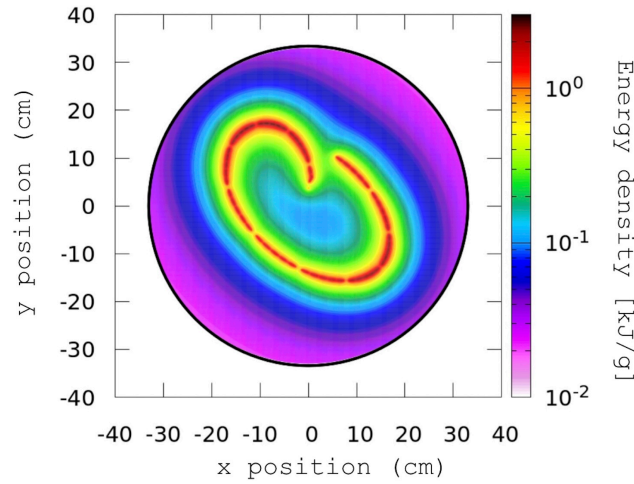


Figure 4.3: Transverse energy deposition density in the low-density graphite segment of the LHC dump core at the section where the maximum energy deposition is achieved (at a depth of about 270 cm from the upstream window). The results correspond to a beam energy of 7 TeV and a bunch intensity of $1.8 \cdot 10^{11}$ protons [20].

4.2 Composition of the Target Dump External

The TDE cores are composed of graphitic materials that are divided into three main sections with the aim of locally optimizing the peak energy load achieved at each point. This is done by modifying the density and length of the different segments. Figure 4.4 shows a cross-section of the TDE cores.

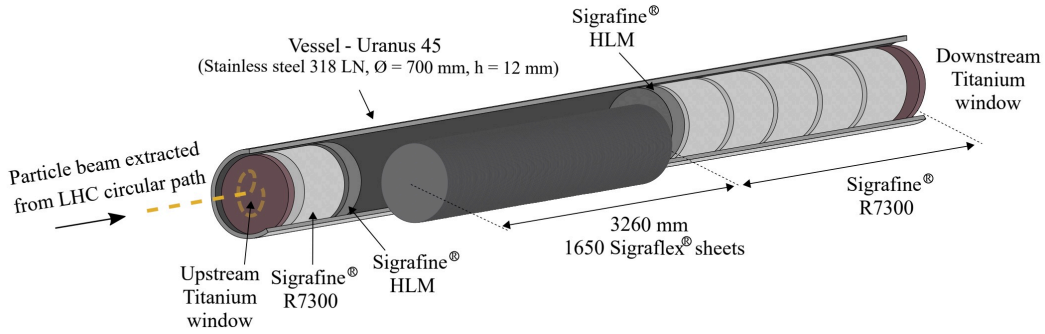


Figure 4.4: Cross section of the Target Dump External showing the core of high- and low density segments [34].

From the left, the beam passes through a Titanium window. The beam travels 50 cm in Nitrogen vacuum before impacting the first high-density block made of 0.7 m of 1.73 g/cm^3 graphite (commercial name SIGRAFINE®, SGL grade R73300) [20] [34]. The block functions as a particle shower initiator that, due to its high density, has a great number of interactions between the beam and the material. The low-density sector is composed of 3.26 m of 1.2 g/cm^3 2 mm thick expanded graphite sheets (commercial name SGL SIGRAFLEX® L20012C [19]). The reason for lowering the density of this sector is to diffuse the particle shower and thereby the energy density over the length of the sector, to reduce the peak energy deposition and loads in the core material. The peak energy deposition is in the low-density sector and occurs at a depth of approximately 270 cm from the upstream window. As seen in figure 4.5, in nominal HiLumi operation, the temperature at the peak will reach 1800°C .

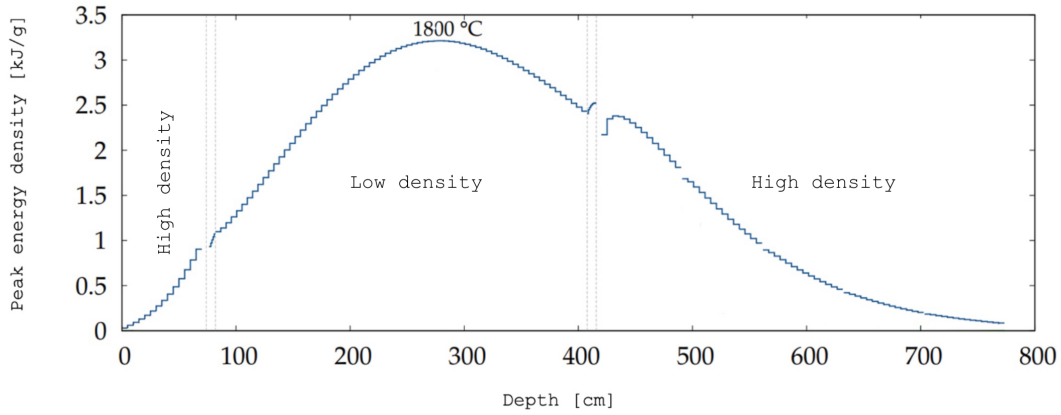


Figure 4.5: Maximum temperature increase at each longitudinal position in the core. The maximum temperature reached is 1800 °C approximately at a 270 cm depth [20].

The SIGRAFLEX® sheets are kept in place by two graphite plates that are aligned with circumferential clips in the vessel. The downstream part of the TDEs consists of five SIGRAFINE® blocks with the same length and density as the first high-density block. These blocks ensure to maximize the energy absorption remaining from the particle shower generated in the low-density sector [20]. As is the case for many other beam-intercepting devices, graphitic materials have been selected as optimal candidates for core materials due to their high thermal and electrical conductivity, conformability, capability of dissipating energy, chemical resistance up to and beyond 2500 °C and low gas permeability [33]. A table of materials in the TDEs can be seen in table 4.1 and a breakdown of the energy deposition in each element is shown in table 4.2.

Part	Material	Density [kg/m ³]	Thermal conductivity [W/(m °C)]	Yield strength [MPa]	Young's module [GPa]
Vessel	SS318LN	7 800	14.1	480	195
Flanges	SS318LN	7 800	14.1	480	195
Windows	Titanium gr. 23, 5	4 430	7.2	834	110
HD blocks	SIGRAFINE R7300	1 730	70.0	38	85
LD blocks	SIGRAFLEX 1.2	1 200	238.0	-	-
Plates	SIGRAFINE HLM	1 730	160	-	-

Table 4.1: Materials used in the Target Dump Externals [21], [27].

Element	Total deposited energy: 432.6 MJ (82%)	
Upstream window	0.001%	0.002 MJ
High-density graphite	49.5%	214.4 MJ
Low-density graphite	42.4%	183.3 MJ
Graphite plates	2.6%	11.3 MJ
Downstream window	0.06%	0.3 MJ
Vessel	5.4%	23.4 MJ

Table 4.2: Energy deposition in TDEs by element [21].

The high-density blocks are shrink-fitted into the vessel. This is to achieve a strong contact pressure, which ensures both the mechanical coupling of the elements and good thermal conductance. A high

value of this parameter is needed to effectively transfer the energy deposited in the dump from the core through the vessel and to the air cooling system. The low-density sector is not shrink-fitted due to the nature of the sheet composition. The vessel has a thickness of 12 mm and is made of 318LN stainless steel (EN 1.4462, also known by the commercial name Uranus 45). The upstream and downstream windows that are bolted to the 318LN flanges are made of Titanium grade 23 and 5 respectively. Inside the vessel is a 0.02 MPa Nitrogen pressure to prevent oxidation of graphite at high temperatures [21].

The TDEs rest on two stainless steel cables that are led through cable guides and into a cradle-like support system. The configuration was one of the upgrades that were implemented for Run 3 operation to allow the TDEs to vibrate and move in a more unconstrained fashion upon beam impact. This feature was sought to prevent the previous more rigid support system from giving rise to excessive constraints. The configuration is shown in figure 4.6 [21].

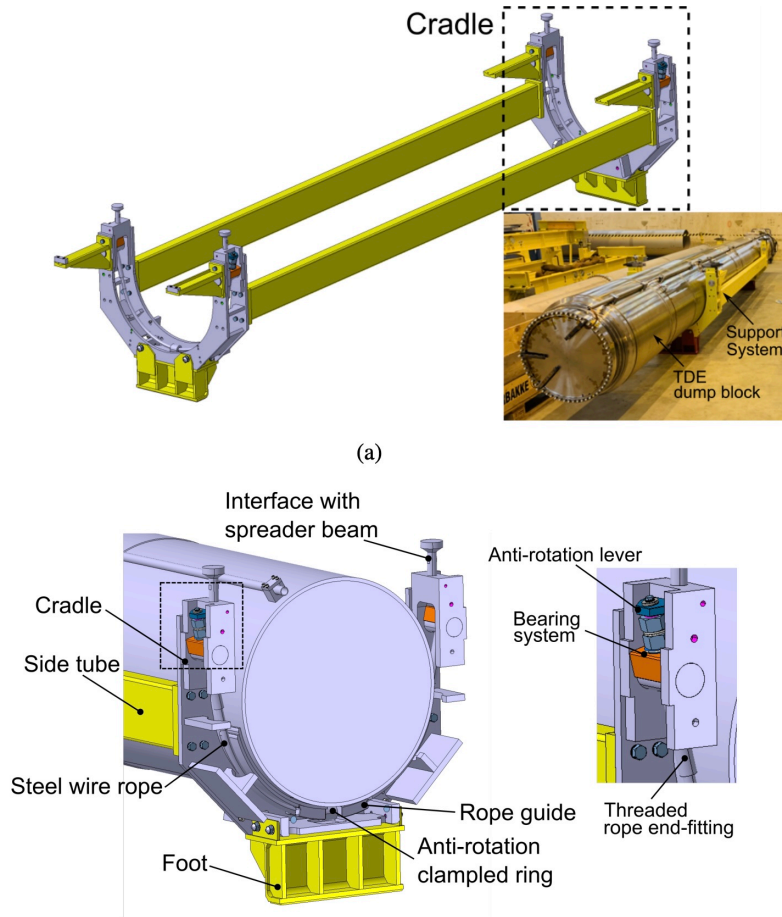


Figure 4.6: Illustration of the TDE support frame and mounting system [20]. The dump is resting on two steel wire ropes that allow the dump to vibrate and move freely upon impact of the beam.

The design of the TDEs is the result of an iterative engineering process over several generations of dumps that involved experimental work of different candidate materials and simulations in FLUKA to simulate the particle shower and generate the energy deposition map. Ansys® and LS-Dyna® finite-element codes are then used to compute the thermomechanical response of the dump core [20].

4.3 Response Upon Beam Impact

During Run 2 several measurement tools were installed in the beam dump cavern to monitor the behaviour of the beam dumps upon beam impact. Interferometers were installed to measure the displacement of the beam dumps and to benchmark with simulations. The data post-processing of these measurements provides a good understanding of the behaviour of the beam dump and gives an indication of the most suitable tools to assess the engineering challenges. The response can be decomposed as the superposition of a fast response and a slow response, as shown in figure 4.7 [21]. During the fast response, strong vibratory phenomena occur as a consequence of the almost instantaneous temperature rise in the vessel.

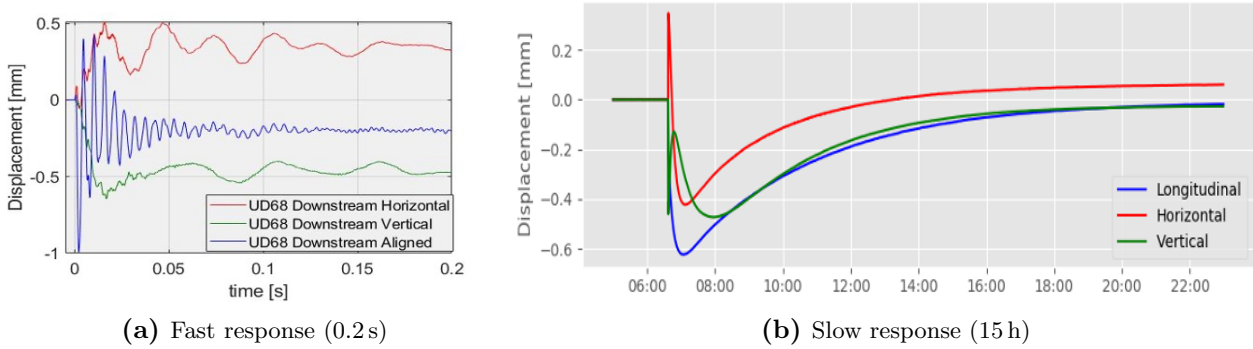


Figure 4.7: Displacements of the Target Dump Externals measured with interferometers

Upon beam impact, the TDEs exhibit fast oscillatory movement / vibration (figure 4.7a). This movement, which occurs as a result of the almost instantaneous deposition of energy in the vessel, can be better understood by decomposing it as the superposition of the eigenmodes of the dump core. The fastest oscillations seen in figure 4.7a are produced along the longitudinal direction at 190 Hz approximately (UD68 Downstream Aligned in figure 4.7a). Vertical and horizontal oscillations (37 Hz) are related to flexural modes. The oscillation amplitudes depend on the beam energy and are rapidly dissipated over a time frame of about 200 ms. This behaviour is referred to as fast movement and is the source of most of the engineering challenges that were found during Run 2. The high-frequency vibrations, in fact, generate accelerations in the dump components of up to 2000 g [20].

A slow and quasi-static response is then caused by the progressive heat transfer of the energy deposited in the core towards the vessel 4.7b. This is then followed by a cooldown process that can take more than 18 hours depending on the beam intensity.

	Time range	Assessment tool (software)	Details
Fast movement	25 ms	LS-Dyna [®]	Vibration due to energy deposition
Slow movement	12 h	ANSYS Mechanical [®]	Thermal expansion of the dump in all directions

Table 4.3: Categorization of TDE response upon beam impact and tool used for assessment.

As a result of the observations, the assessment of the design for the Run 3 spares is focusing on the fast movement of the dump using an explicit Finite Element Method in LS-Dyna. A summary of the behaviour of TDEs can be seen in table 4.3. A more detailed description of LS-Dyna and the assessment approach will be outlined in the following chapter.

The currently operational dumps are the Run 2 dumps installed in 2015 that underwent several

upgrades before Run 3 during the Long Shutdown 2 in 2018. While the upgrades aimed to improve the capability of the dumps, the dumps are not designed to withstand beams of Run 3 nominal energy. In case one of the dumps fails during Run 3 no current spare dumps are available for replacement, highlighting the need for manufacturing of Run 3 spares.

4.4 Modifications for the Run 3 Spares

The two modifications for the Run 3 spares under evaluation in this report are a new design for the downstream flange and a different weld configuration of the vessel.

Figure 4.8 shows an overview of the different versions of the downstream flange through the different LHC Runs. The version labelled 2018 is the one that is implemented in the currently operational dumps. It corresponds to the original design with a welded extension. During Run 2, in fact, it was found that the chain mounting system of the window to the flange was leaking. Due to the inaccessibility of the dumps, it was preferred to weld a new extension on the existing flange that allowed the bolting of the upstream window to the extension [21].

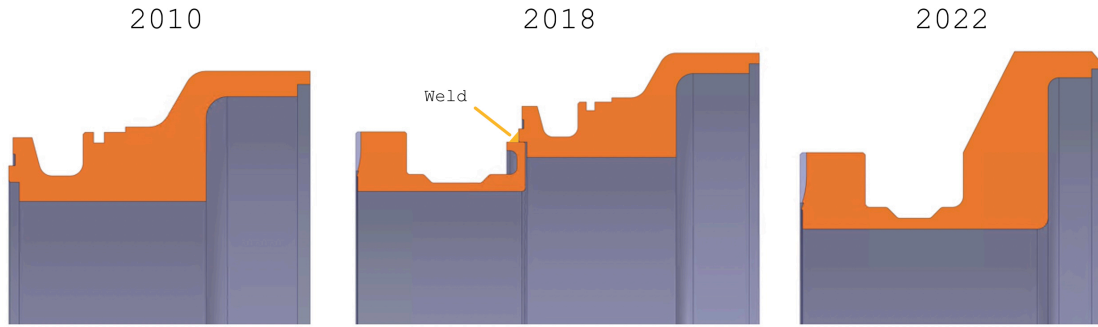


Figure 4.8: Evolution of the downstream flange design and the proposed design for Run 3 spares (2022). On their right boundary, the flanges are welded to the vessel. A Titanium window is bolted to their left face.

The proposed design for Run 3 spares (2022) is a simplification of the 2018 design that includes the bolting feature. This is possible since the Run 3 spares will be built as new devices, unlike the currently operational dumps. The design has a different geometry, which must be validated to ensure safe operation throughout its life.

The current weld configuration can be seen in figure 4.9. 9 circumferential welds along the high-density sector and 2 longitudinal welds along the low-density sector enclose the vessel. The reasoning behind this configuration is that the high-density sectors are exposed to high stress levels due to the shrink-fitting of the core materials and therefore a longitudinal weld in these areas would be exposed to high static stress even before applying the loads upon beam impact. In the low-density sector, the core is not shrink-fitted due to the nature of the sheets and therefore two vessel segments are welded together in the longitudinal direction. The welds are partial penetration butt welds through the 12 mm thickness of the vessel with a depth of 6.65 mm. A cross-section of the weld cap can be seen in figure 4.11. The vessel segments have a male-female configuration, which makes the vessel segments self-aligning upon assembly and welding.

The proposed weld configuration can be seen in figure 4.10. 12 circumferential welds along the low- and high-density sectors join and enclose the vessel. This weld configuration is adopted mainly

to streamline the manufacturing process. The circumferential welds are performed after the shrink-fitting of the high-density blocks, resulting in the welds not being exposed to the stress caused by the shrink-fitting.

Both the current and the proposed weld configuration have not previously been assessed and due to increased energy deposition in the TDEs, an evaluation of the weld strength upon beam impact must be performed.

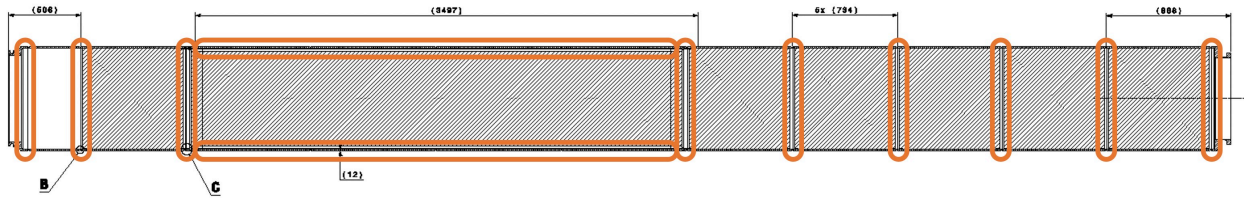


Figure 4.9: Current weld configuration in the TDEs. Nine circumferential welds in the high-density sectors and two longitudinal welds along the low-density sectors. Welds are highlighted in orange.

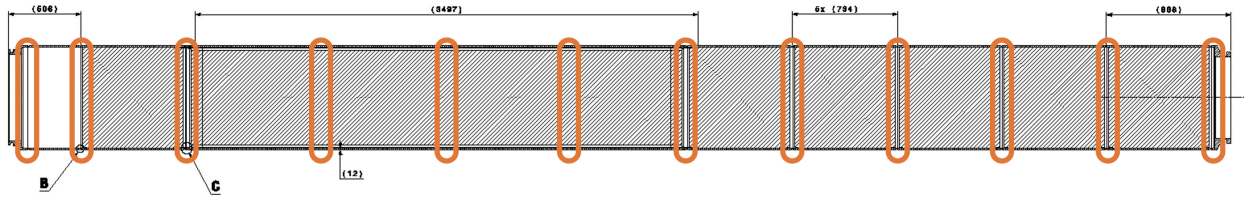


Figure 4.10: Proposed weld configuration for construction of the run 3 TDE spares. Twelve circumferential welds along the dump. Welds are highlighted in orange.

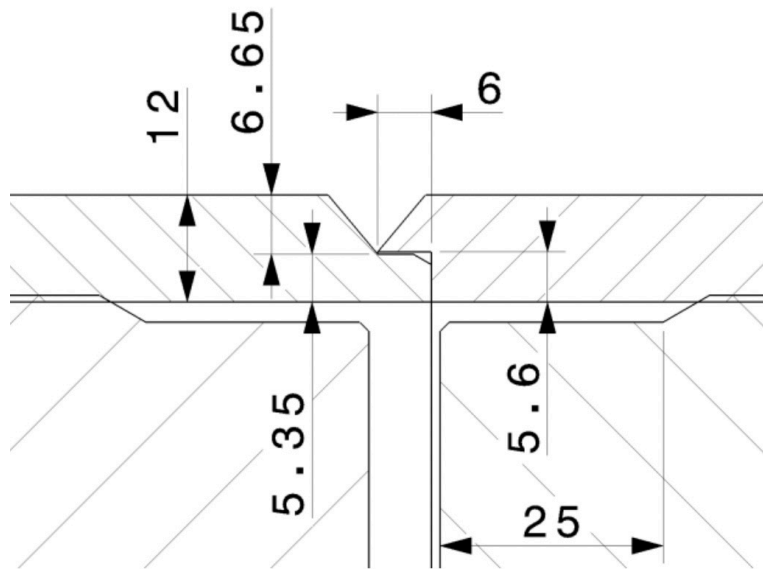


Figure 4.11: Cross section of Target Dump External highlighting the current weld configuration. The weld is a partial penetration weld with a depth of 6.65 mm allowing a male-female design to ease assembly and welding.

The new flange design and the weld configuration are critical aspects and possible sources of structural failure of the TDEs. These aspects are essential to ensure a satisfying design with a reasonable safety

CHAPTER 4. THE TARGET DUMP EXTERNALS FOR THE LARGE HADRON COLLIDER

factor before the manufacturing of the Run 3 spares begins in 2023. The following chapter will give an overview of LS-Dyna[®], which has been the primary tool for assessing these matters.

Finite Element Method

The Finite Element Analysis (FEA) using the Finite Element Method (FEM) is a numerical technique that uses computational power to calculate approximate solutions. It is widely used in all major engineering industries to reduce the number of prototypes needed to be developed to converge to a satisfactory engineering design and to gain a better understanding of the behaviour of the component under load and how to assess it [24].

The FEM can be used to assess a wide range of different problems involving static, dynamic, buckling, and model analysis of solid mechanics as well as fluid and thermal problems. The goal of an FEA in dynamic solid mechanics is to calculate stresses, strains, and displacements within the component to evaluate weak points and identify possible modifications to strengthen them [24].

Internal stresses develop within a body in such a way as to maintain equilibrium over any volume of the body. Therefore, the field variables (stress, strain, and displacement) can be calculated by applying the concept of equilibrium. The FEM does this by splitting the body into a number of small elements that are connected at nodes. This process is called discretisation, and the collection of nodes and elements is called the mesh [31].

5.1 Implicit vs. Explicit Algorithms

An explicit transient thermomechanical coupled analysis is used to compute the dynamic response of the TDEs. The underlying equation solved by the FEM in the transient analysis can be schematized as a dynamic equilibrium:

$$[M] \cdot \ddot{U} + [C] \cdot \dot{U} + [K] \cdot U = F(t) \quad (5.1)$$

Where:

- $[M]$ is the mass matrix of the component.
- $[C]$ is the dampening matrix.

- $[K]$ is the stiffness matrix.
- $\ddot{U}$, $\dot{U}$ and U are the vectors of nodal acceleration, velocity and displacement respectively
- $F(t)$ is the external forces vector.

The objective is to solve the displacement vector U since the matrices are material-dependent quantities and the external forces are known or applied. Once U is obtained, the stress can be derived by computing the strain.

The displacement U of the equation of motion (5.1) can be solved with a variety of integration methods that can belong to the following two types of algorithms:

- Implicit algorithms

The solution at time step $n + 1$ is obtained by knowing the solution at time step n and the conditions imposed at step $n + 1$. If certain stability conditions are satisfied, these algorithms are *unconditionally* stable. This means that the method is stable regardless of the time size adopted for the time step. An appropriate time step size is evaluated through the concept of convergence. The upside of these algorithms is that the time step size can be large and so the expense of central processing units (CPU) low. The downside is that in order to obtain a solution, the matrices need to be inverted at every time step, which repeatedly requires a substantial amount of memory [35].

- Explicit algorithms

The solution at time step $n + 1$ is obtained by knowing the solution and the conditions imposed at time step n . These algorithms are *conditionally* stable, which means that the stability of the methods depends on the time step size adopted. The upside of these methods is that there is no inversion of matrices, and so the memory usage is low. The downside is that as a consequence of the small time step size, the need for central processing units (CPUs) can increase substantially if longer times need to be simulated [35].

Based on the characteristics of the algorithms, the selection of the appropriate method can be made following the criteria illustrated in figure 5.1. Problems involving high levels of strain rates are assessed using explicit algorithms [16]. The implicit algorithms are selected for static equilibrium and quasi-static responses.

The explicit algorithms are well suited to capture the dynamic response of the TDEs defined in section 4.3. Therefore LS-DYNA, which is a recent ANSYS-acquired explicit solver, is chosen to assess the problem and will be the main simulation tool to obtain the dynamic response of the TDEs upon beam impact.

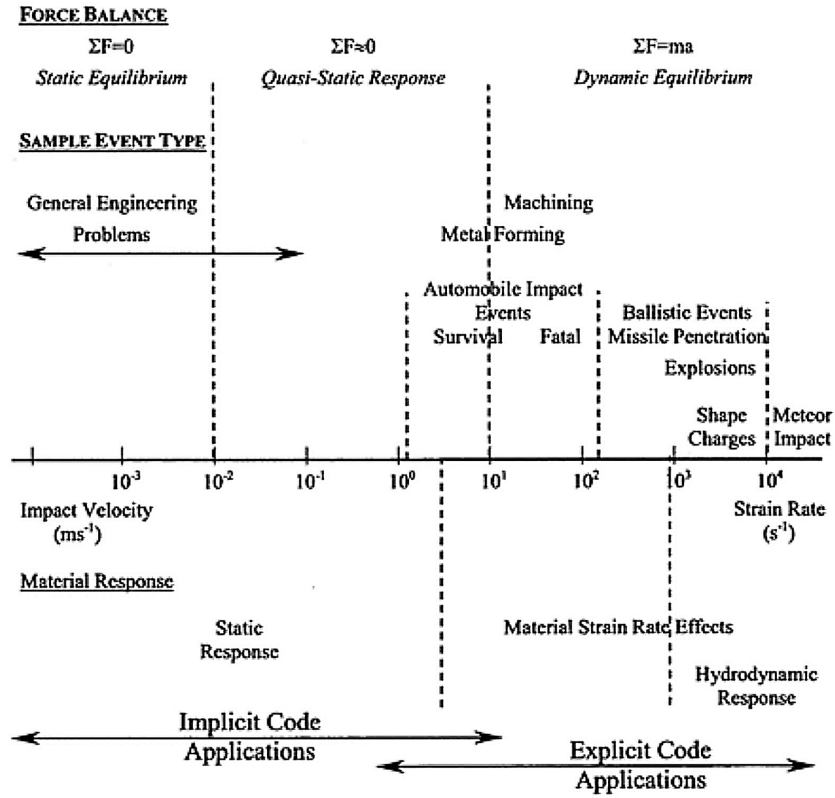


Figure 5.1: Typical strain rates and suggested integration methods for a variety of engineering cases. At low strain rates, implicit methods are suggested due to the unconditional stability allowing large time steps. At high strain rates, explicit methods are suggested to properly capture the behaviour at very low time step sizes without the negative impact of repeated inversion of the stiffness matrix [16].

5.2 Explicit Algorithm Stability Condition

To make the explicit algorithms stable in order to yield accurate results, the Courant-Friedrichs-Lewy (CFL) condition must be satisfied for the convergence of the partial differential equation integration method [16]. The condition expresses that the distance that any information travels during the time step length within the mesh must be lower than the distance between mesh elements. Information from a given cell or mesh element must propagate only to its immediate neighbouring elements [32]. This concept is illustrated in figure 5.2 where the upper time step (red) is too large, resulting in a Courant number greater than 1, yielding an unstable solution that does not correctly capture the behaviour of the information in all elements of the model. The lower time step (blue) illustrates a Courant number that is less than 1, resulting in a correct distribution of information throughout the model. A simple equation to emphasize the concept:

$$\frac{v\Delta t}{\Delta h} < C_{max} = 1 \quad (5.2)$$

Where v is the velocity of propagation of the information, Δt is the time step size, and Δh is the characteristic length of the element. The velocity multiplied by the time step size gives the length the information has travelled during the time step. If the length is longer than the characteristic

length of the element, the Courant number is greater than 1, resulting in instability.

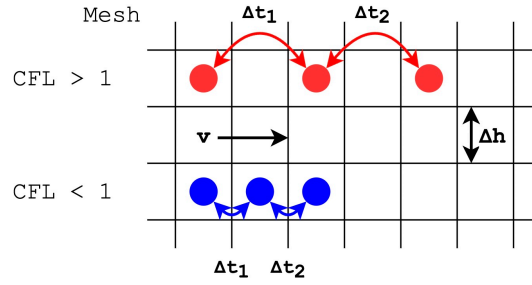


Figure 5.2: Visualisation of the Courant number and the CFL condition on a computational grid.

For explicit simulations in solids, equation 5.2 can be rewritten into the following form:

$$\Delta t_{max} \leq f \cdot \frac{\Delta h_{min}}{c} \quad (5.3)$$

Where Δt_{max} is the maximum time step size possible to ensure stability, f is a safety factor (often 0.9), Δh_{min} is the characteristic length of the smallest element in the model, and c is the wave speed in the material (speed of sound in the material) which can be calculated as follows:

$$c = \sqrt{\frac{E}{\rho}} \quad (5.4)$$

where E is the young modulus and ρ is the density of the material.

The CFL condition is a critical aspect when adopting explicit algorithms. A typical model can easily have time step sizes less than a millionth of a second due to the mesh element size. Therefore, when meshing for explicit analysis one has to be careful. Not only does a refinement of the mesh increase the number of elements and thereby the computational cost, it also reduces the time step size resulting in more time steps over the simulation time which adds another factor to the computational cost.

Simulation of Dynamic Response Upon Beam Impact

The simulation of the dynamic response of the TDEs upon impact of the beam is the basis for the evaluation of the downstream flange and the welds in the vessel. As will be seen in the following, this simulation of the TDEs will be the boundary condition imposed when setting up the submodels for the downstream flange and the welds in the vessel, transferring the resulting local displacements of the TDEs to the submodels and thereby loading and constraining them. The assessment workflow is shown in figure 6.1 and will be referenced for workflow clarification throughout the following chapters.

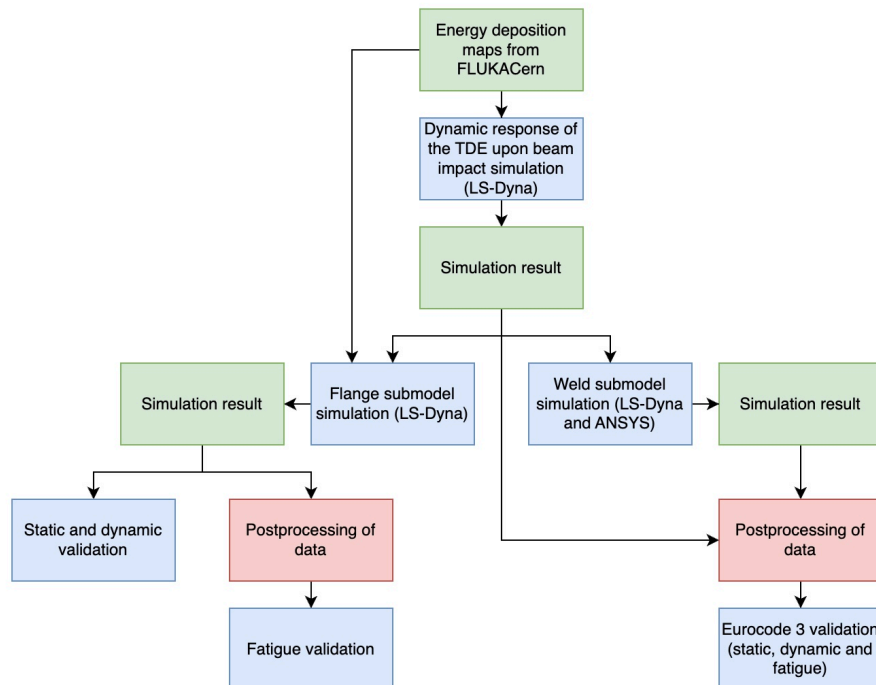


Figure 6.1: Assessment workflow for the downstream flange and the welds in the vessel, describing the simulation flow and dependencies.

As seen in the previous chapter, during Run 3 operation, the TDEs are subject to up to 538 MJ. This triggers a fast dynamic response during which the TDEs vibrate as a result of the sudden

temperature increase in the vessel. This thermal loading is complex because of the multiple physics phenomena at play. As such, the energy deposition in the dump core has to be computed using the FLUKA Monte Carlo Code [10] which will be explained in the following section.

6.1 Energy Density Map Using FLUKA

The FLUKA Code [4] [11] [8] is a general-purpose Monte Carlo code developed at CERN to perform calculations of particle transport and interactions with matter. It was built with the aim of including the best physics models in completeness and precision through a microscopic approach, where each step in the analysis has a sound base in physics. With the comprehensive facilities at CERN, the code has been benchmarked on a single interaction level with experiments for the pursuit of reliability [10].

Figure 6.2 shows a simple FLUKA simulation in which protons with an energy of 150 MeV impact water. The figures on the left are visual interpretations of the particle tracks. From 6.2a to 6.2e the number of simulated protons increases. It can be observed from figure 6.2b to 6.2f that with a higher number of protons impacting the water, the statistical uncertainty of the average energy deposition decreases.

When the particles propagate in matter, they are subject to a series of radiation-matter interaction mechanisms, leading to the following:

- Angular deflections (target atom displacement)
- Energy losses (energy transfer to material, heating up)
- Generation of secondary particles (formation of a radiation shower also called particle shower)
- Production of residual nuclei, possibly radioactive (emission of prompt/delayed radiation)

These mechanisms, as seen on the particle tracks in figure 6.2 a, c and e, are built into the code and occur based on the physics phenomena and related probability distributions. An in-depth explanation of the nuclear interaction mechanisms can be found in [10], but are outside the scope of this thesis.

For the purposes of the current work, the key output of interest of the FLUKA simulations is the energy deposition that is caused on average by each proton impacting the dump core. It consists of a 3D energy density map (with units of GeV/cm^3 per primary particle). It has an elemental structure similar to a mesh that contains the locally averaged energy deposition [30]. FLUKA-generated energy density maps do not include information about the time domain. This means that the beam that is swept over the cross-section of the dump (figure 4.3) during the $86 \mu\text{s}$ impact is simplified considering the accumulated energy deposition during the entire impact period of the beam. The intensity is then linearly ramped up in the LS-DYNA[®] simulation. This is considered a reasonable simplification for the simulation of the response of the global model. Indeed, the beam sweep occurs much faster than the eigenmodes that this process excites. Simulations with the full-time structure of the beam sweep are, however, necessary to accurately simulate the response of the TDE windows.

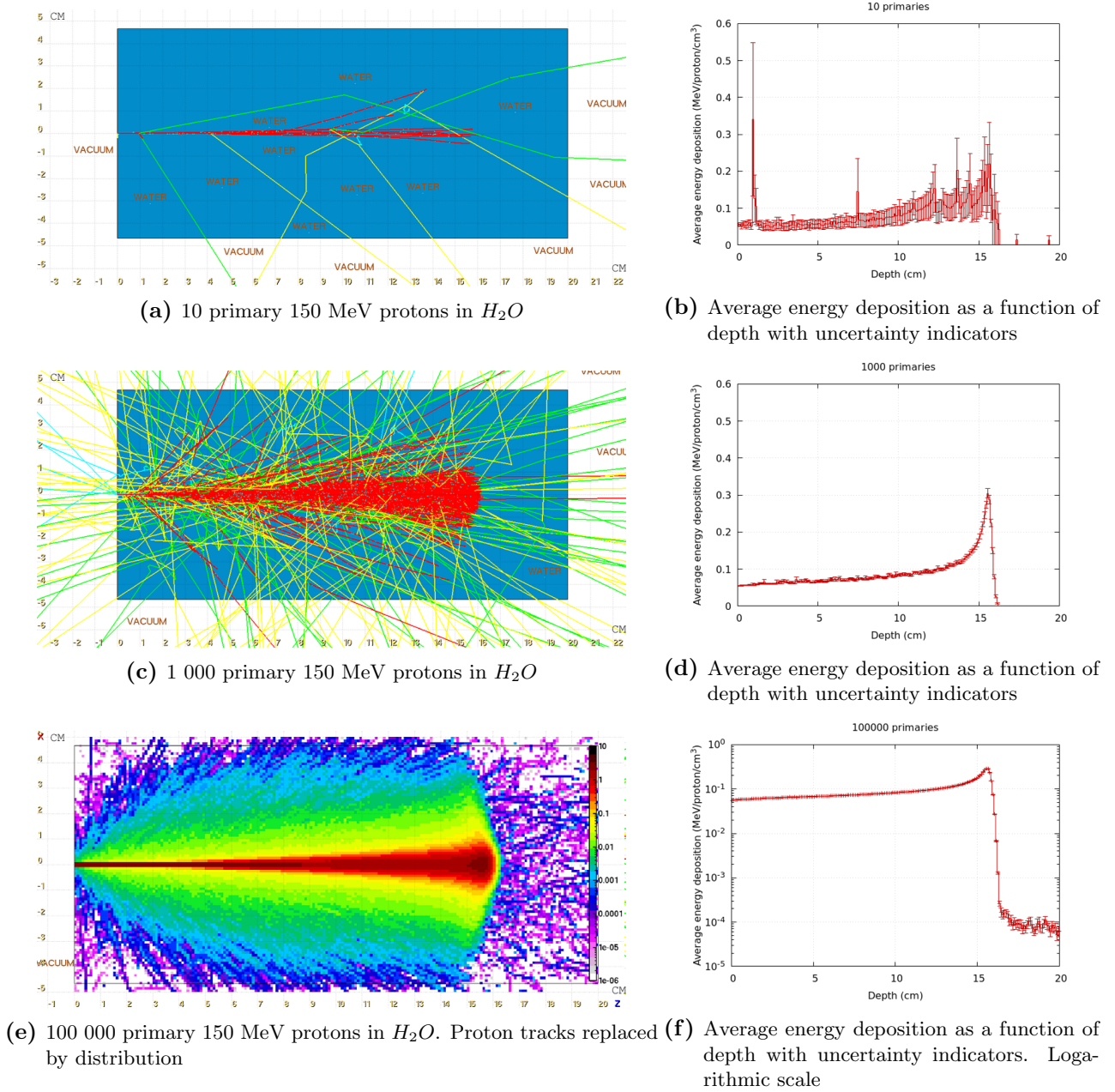


Figure 6.2: Illustration of Monte Carlo energy deposition simulation of 150 MeV protons in water. From a to f the number of Monte Carlo iterations is increased and thereby the uncertainty decreases. In figures a, c, and e red lines illustrate protons, yellow lines photons, light blue lines electrons and green lines neutrons [30].

6.2 Simulation of Shrink Fitting

The following section will cover the setup of the simulation of the TDEs including a description of the workflow and considerations involved. The process can be seen in figure 6.3. The majority of the setup is performed in Ansys (geometry preparation, meshing, contact definition, loads and constraints) due to an intuitive GUI and powerful geometry and meshing tools. The implementation of the energy density map is based on an Ansys extension developed internally using the Ansys Parametric Design Language (APDL). APDL allows users to write custom scripts to interact with

the solver [37]. An LS-Dyna extension within Ansys enables the possibility of writing the LS-Dyna input file from Ansys containing everything from mesh to loads, which can then be run using the explicit code in LS-Dyna.

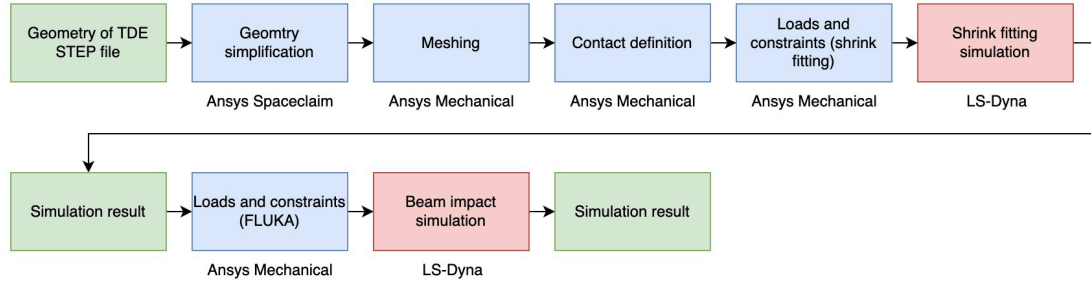


Figure 6.3: Workflow for the simulation of the dynamic response of the TDE upon beam impact.

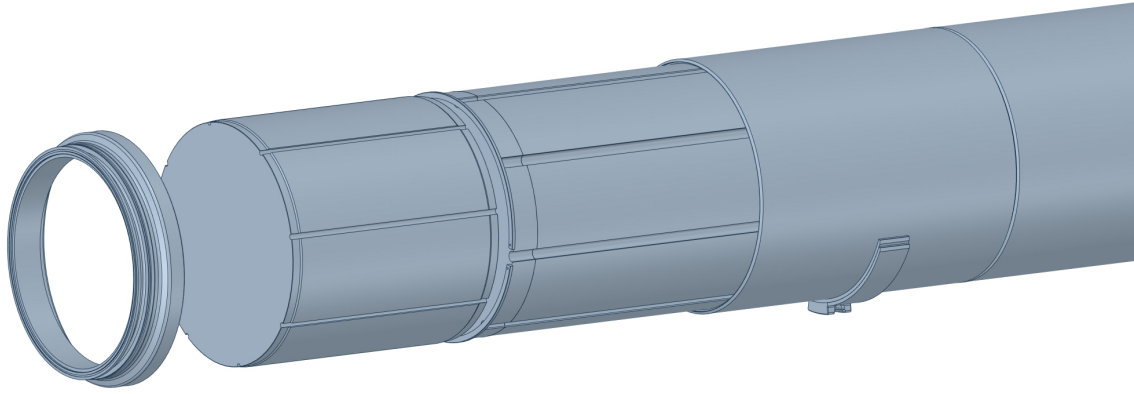
The simulation is a multistep analysis including a simulation of the shrink fitting of the high-density core segments in the vessel, which is then imposed in the second simulation of the beam impact as a starting condition. This means that the vessel before beam impact is pre-stressed. It is important to include the shrink fitting in the beam impact simulation because the core elements due to the shrink fitting contribute to the structural stiffness of the vessel, reducing vibrations and, thereby, the dynamic response. Furthermore, the shrink fitting contributes to non-negligible stress levels in the high-density vessel segments of the TDE.

The shrink-fitting process does not have a dynamic nature. It is obtained as a consequence of slow heating and subsequent cool down of the vessel subsection contained graphite block. As such, a good result can be obtained using standardized shrink-fitting hand calculations following ANSI B4.1 [26] or using an implicit FE method in Ansys Mechanical[®]. However, to streamline the simulation workflow, it was chosen to perform the shrink-fitting analysis in LS-Dyna[®] easing the transfer of the pre-stressed state into the beam impact simulation. To ensure the validity of the shrink-fitting result obtained in LS-Dyna[®], the result is benchmarked against hand calculations and implicit simulation in Ansys Mechanical[®]. These results and differences will be presented at the end of this chapter.

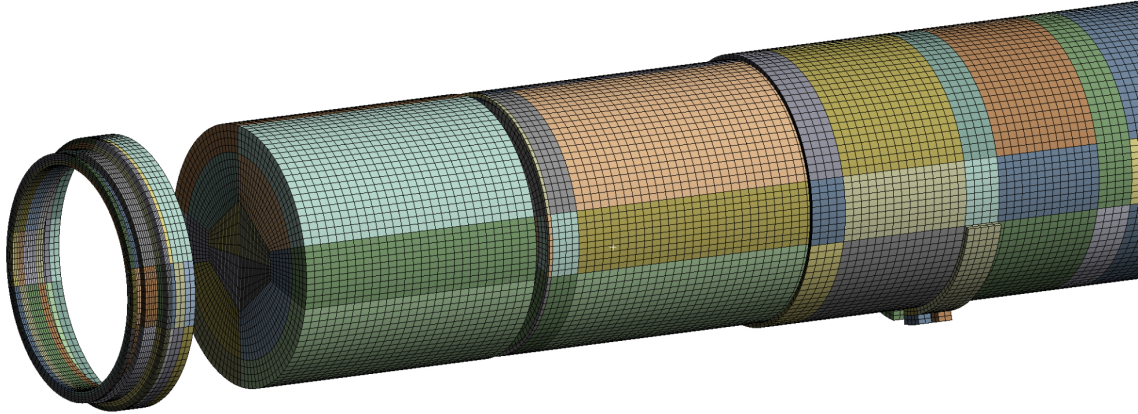
The geometry is simplified to ease the meshing process. Simplification involves the removal of fillets, welds, holes, and other features that do not affect the overall stiffness of the system. An in-depth description of the defeating of the TDE can be found in A.2. A.2 also includes various examples of meshing, mesh metrics, and CFL timestep analysis.

Due to small features such as the 12 mm thickness of the vessel, the mesh must be relatively fine to properly capture the geometry of the TDE. The global mesh element size is set to 2 cm, and 33 local mesh settings (method and size) are applied to assist the meshing algorithm. An example of the meshing result can be seen in figure 6.4. The final mesh consists of 785 000 linear hexahedral elements and 700 000 nodes. The TDE is meshed according to the guidelines found in A.3.

The shrink fitting process is performed considering a diametral interference of 0.75 mm. As such, the outer diameter of the graphite blocks is 700 mm and the inner diameter of the vessel is 698.5 mm. The contact between the core segments and the vessel is a penalty-based surface-to-surface interference frictional contact that is applied over a 0.5 ms duration. Since the contacts are applied over a short period of time, the vessel vibrates. A special LS-Dyna[®] preload simulation tool known as Dynamic Relaxation (DR) is implemented, which applies heavy damping to the system to level out the stress



(a) The original geometry



(b) The upstream part of the TDE meshed after geometry modifications

Figure 6.4: The geometry modifications and meshing of the upstream part of the TDE. From the figures, the upstream window and three vessel segments are hidden, exposing the high- and low-density blocks.

oscillations and stabilize the solution. Furthermore, a slight pressure of 0.02 MPa is applied to all the inner surfaces of the TDEs to simulate the Nitrogen pressure inside the vessel. A sensitivity analysis of the load curve that scales the contact stiffness over the 0.5 ms simulation period allowed the possibility of obtaining a result consistent with the implicit analysis and hand calculations.

The simulation is run on 64 cores on a cluster using Massively Parallel Processing (MPP). MPP uses a message-passing protocol to exchange information between cores over a network. To do so, it performs a domain decomposition of the model to distribute it among the cores [22]. The use of MPP makes it possible to solve the simulation on a large number of cores instead of being limited to typically 8 cores on the local PC, ultimately reducing the simulation time significantly. Another advantage is that running simulations on a cluster makes it possible to run several simulations in parallel, allowing sensitivity studies to be run simultaneously.

6.2.1 Shrink Fitting Results

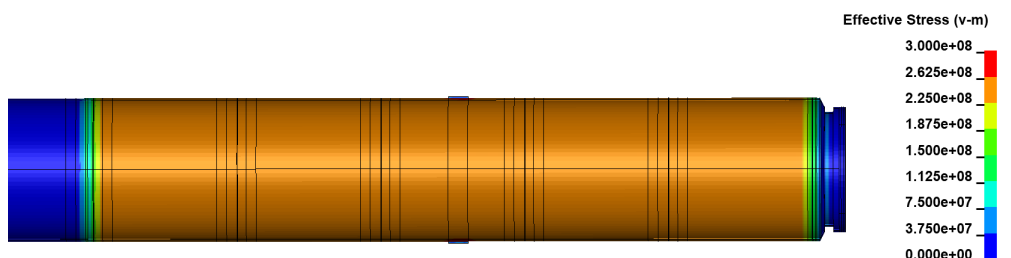
The result of the shrink-fitting of the high-density core elements can be seen in figure 6.5 clearly highlighting the pre-stressed high-density sectors at approximately 261 MPa. Hand calculations following ANSI / ASME B4.1 showed a von Mises stress level of 266 MPa and the implicit simulation in Ansys Mechanical[®] resulted in a von Mises stress of 256 MPa. The implicit analysis simulated a

single high-density block shrink-fitted in a vessel segment and can be seen in figure 6.6. The documentation and description of hand calculations can be found in A.4. The shrink-fitting results along with their differences can be seen in table 6.1.

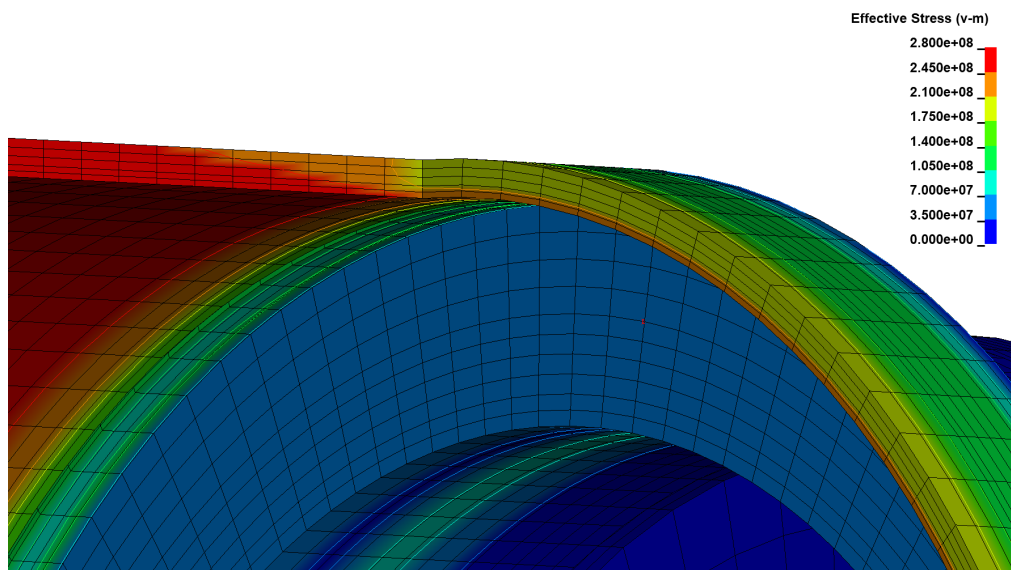
Tool	Von Mises stress in vessel	Difference
	[MPa]	%
LS-Dyna explicit simulation using DR	261	-
Ansys Mechanical implicit simulation	256	1.93
Hand calculations according to ASME/ANSI B 4.1	266	1.89

Table 6.1: Von Mises stress in TDE vessel due to shrink-fitting of core elements.

The results are consistent between all methods within a tolerance of 2 % which is considered satisfactory. The state of the model is transferred to the set-up of the beam impact simulation as a starting condition. The setup of the beam impact simulation is described in the following chapter.



(a) Downstream part of TDE containing 5 shrink-fitted high-density blocks.



(b) Cross section of the vessel at the downstream flange transition. The high-density block is hidden.

Figure 6.5: Shrink-fitting results in LS-Dyna.

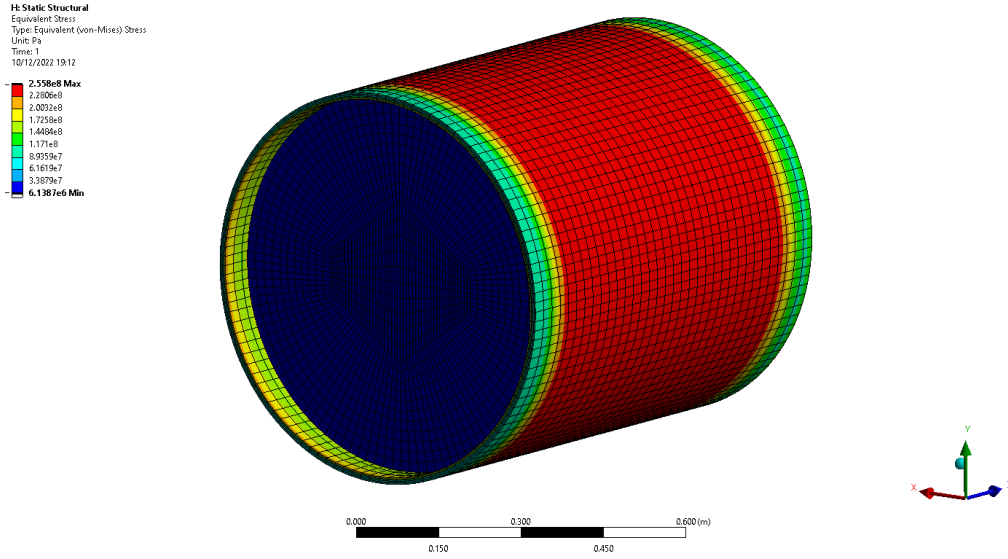


Figure 6.6: Shrink fitting analysis of a single high-density block in a vessel segment.

6.3 Simulation of Beam Impact

To simulate the impact of the beam on the TDEs, an energy density map is generated using the FLUKA Monte Carlo code [10]. The energy density map together with the pre-stressed state as a result of the shrink fitting are the loads applied in the simulation. To capture the dependencies of energy deposition and dynamic response, a transient weakly coupled thermomechanical simulation is adopted.

Due to the short impact duration of $86\mu\text{s}$ the only energy applied from the particle shower is the energy deposited in the vessel. Such a load simplification was first adopted in the assessment of the Run 3 operational dumps [21] and was shown to yield satisfactory results. This is not the case, however, for the assessment of the slow movement described in chapter 4.3. For this case, the energy deposition in both the vessel and the core has to be included because the expansion of the dump is driven by thermal conductivity between the core, vessel, and the cooling system's capability of cooling the dump via convection on the outer surface of the vessel. However, as stated previously, the assessment of the slow movement is beyond the scope of this report.

The shrink-fitting result is imposed on the model as a starting condition, meaning that the stress results during the beam impact simulation are the accumulated effect of the shrink-fitting stress and the stress due to the dynamic response upon beam impact. The implementation of the energy density map is performed using an Ansys[®] extension developed within the section. The extension allows one to import the energy density map into Ansys[®], ensuring that it is properly aligned with the geometry (vessel). It automatically extrapolates the energy density from the FLUKA-generated map and interpolates it into the Ansys[®] mesh following a variety of user-defined criteria that can be selected to best fit the simulation case. For this operation to work correctly, it is especially important to ensure that the sizing of the energy density map is within an acceptable tolerance with respect to the model mesh. If the difference between the two meshes is substantial, the interpolation will be inaccurate, resulting in an incorrect representation of the area-dependent depositions. Figure 6.7 illustrates a simplified 2D example of interpolation between two inconsistent meshes. An average energy deposition is assigned to every square in the source energy density map. The interpolation from a single element in the energy density map to four interfering elements of the model mesh is

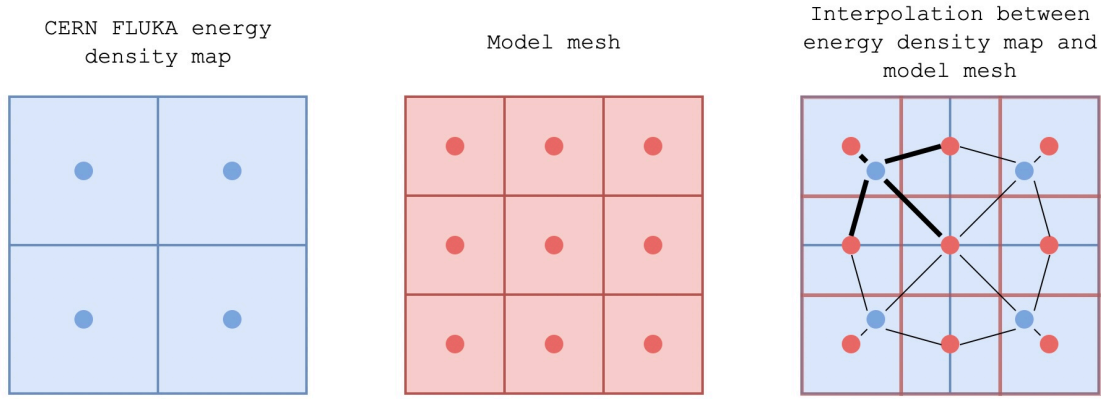


Figure 6.7: Left: energy density map from Monte Carlo simulations structured in a 2D grid. Center: Model meshed ready for interpolation. Right: Illustration of the interpolation between the energy density map and the model mesh. In reality, interpolation is performed between 3D meshes.

highlighted with thick lines in figure 6.7.

To verify a correct interpolation of 785 000 elements, two checks are performed. One check is to compare the peak energy density and the location where it occurs in the FLUKA energy density map with those occurring after interpolation in the Ansys[®] model. The second check is to compare the total energy deposition of the energy input with the one occurring in the Ansys[®] model after interpolation. The results of these checks can be seen in table 6.2 and are considered acceptable. Notice that for peak energy deposition, there is a substantial difference in flanges and windows that is caused by large differences in mesh size between FLUKA and Ansys[®] models. However, since 97 % of the total energy is deposited in the vessel, these inaccuracies are considered negligible. For the vessel, there is a 0.8 % difference in peak energy density and a 0.04 % difference in total energy between the FLUKA energy density map and the Ansys[®] model after interpolation. This result is considered satisfactory. The comparison of total energy can be seen in table 6.2 and peak energy in table 6.3.

	Energy in FLUKA [KJ]	Energy in Ansys [®] [kJ]	Diff. %	Per. of total %
Upstream window	4.99	4.54	-8.9	0.02
Upstream flange	25.0	26.1	4.2	0.11
Vessel	23828	23818	-0.04	97.3
Downstream flange	411	400	-2.6	1.63
Downstream window	226	233	3.2	0.95
Total	24495	24482	-0.05	

Table 6.2: Comparison of total energy deposition in the FLUKA file and after interpolation in Ansys[®].

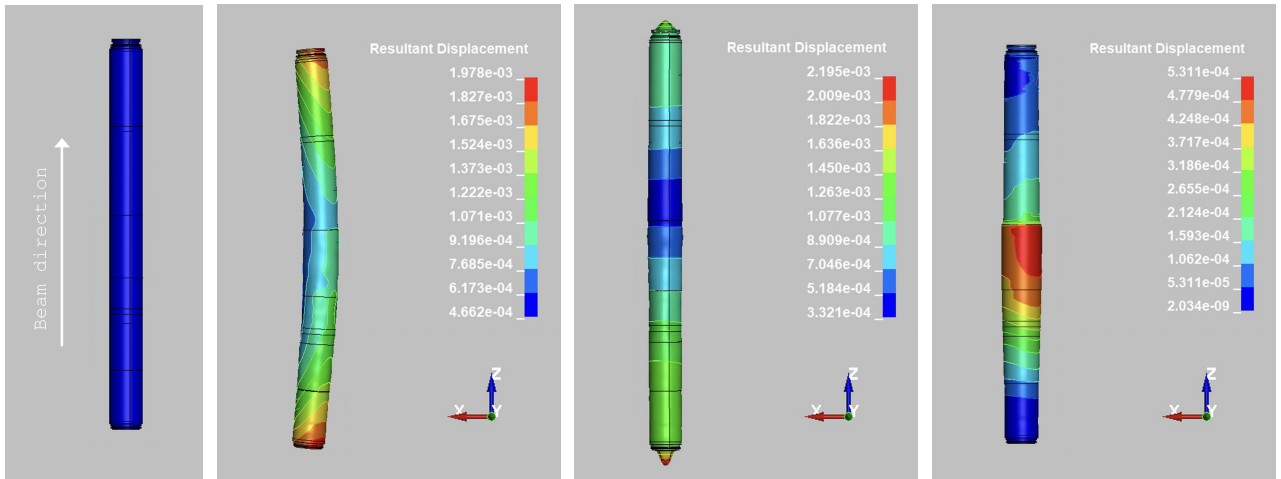
No artificial dampening is applied, and because the support frame and cables holding the dump are not included, the dump can vibrate freely. Especially when considering fatigue for the welds and flange, this is a conservative approach since the undamped vibrations will result in an increased number of high-amplitude stress oscillations. The initial result of interest is the maximum stress in the flange and welds in relation to the yield strength of the material. For fatigue assessments later in this thesis, a measured damping factor based on LDV measurements will be applied to gain a more realistic interpretation of the fatigue aspect.

	Peak energy in FLUKA [W/mm ³]	Peak energy in Ansys® [W/mm ³]	Diff. %
Upstream window	893	396	-55.7
Upstream flange	259	226	-12.4
Vessel	4728	4686	-0.88
Downstream flange	1535	1447	-5.7
Downstream window	4847	2911	-39.9

Table 6.3: Comparison of peak energy in the FLUKA file and after interpolation in Ansys®.

6.3.1 Beam Impact Results

As previously described, upon beam impact, the dump cores experience a dynamic response that can be decomposed as a superposition of mode shapes. As seen in figure 6.8 the main modes that are excited are the flexural modes at 39 Hz, a longitudinal one at 190 Hz and a radial (or *breathing*) mode at 2 205 Hz. The vibrations result in a complex multi-axial stress state.



(a) Model before beam impact. (b) Horizontal bending at 0.019 s. Displacement in m. (c) Longitudinal expansion at 0.002 s. Displacement in m. (d) Breathing at 0.0002 s. Displacement in m.

Figure 6.8: Quasi-static movement of the TDEs upon beam impact. Characterization of different modes. Illustrations scaled by a factor of 150. The color shows displacements.

The maximum von Mises equivalent stress reached in the vessel during the simulation time is in the order of 300 MPa and is located at the level of third and fourth high-density blocks on the top right side of the vessel. It should also be noted that the angular position of the location of maximum Von Mises equivalent stress is well aligned with the position of peak energy density induced by the beam sweep. A deformation scaled result of the TDE can be seen in figure 6.9 highlighting the mode shapes.

In figure 6.10 the simulated displacements of the TDE upon beam impact at the downstream flange is seen. The blue line represents the longitudinal expansion of the dump, the red line the horizontal bending and the orange line a product of vertical bending and radial expansion. It can be derived that the amplitude of the longitudinal mode is higher than the other modes resulting in this mode having the highest stress contribution in the dump. Despite considering the length of the TDEs a 2.2 mm displacement is substantial taking the stiffness of 318LN into account.

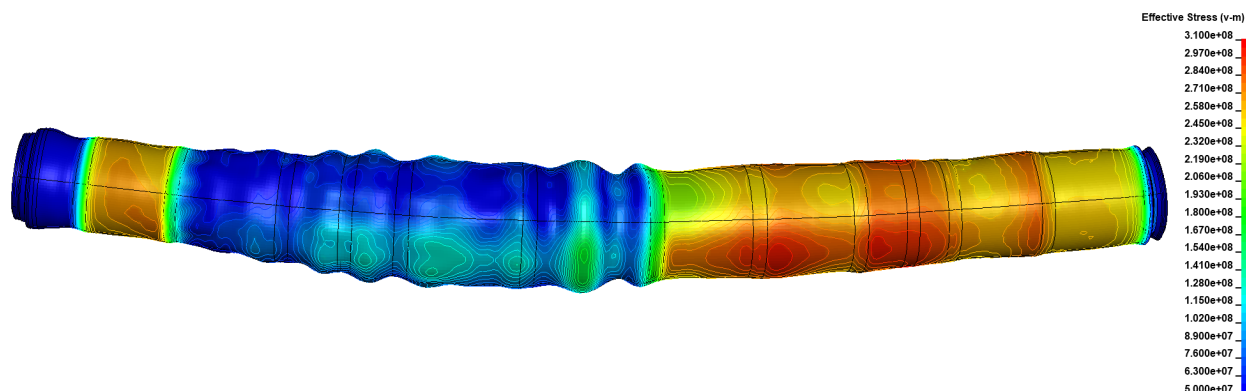


Figure 6.9: TDE response at 5 ms when maximum von Mises stress is reached. Model displacements scaled by a factor of 400 and seen from the top (beam impacting from the left side). The colour shows von Mises stress.

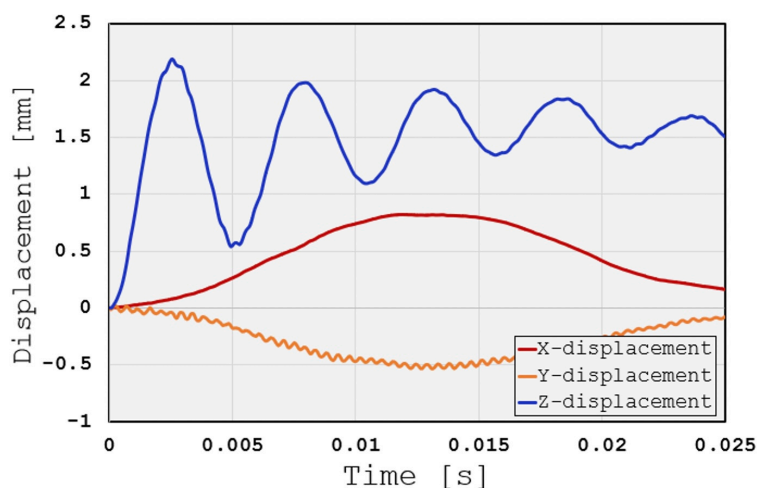


Figure 6.10: Simulated x, y, z displacements of the TDE upon beam impact at the downstream flange (0.025 s).

To gain a sense of the precision of the results, the modal frequencies are compared with the results of a modal characterization experiment described in [21]. Looking at the mode shape characterization experiment in table 6.4 the difference in frequency is 5.2 %, 3.6 % and 6.8 % for the horizontal, longitudinal, and radial modes, respectively, with respect to the simulation.

Result	Horizontal bending Hz	Dif.	Longitudinal expansion Hz	Dif.	Radial expansion Hz	Dif.
LS Dyna simulation	39	-	190	-	2 205	-
Mode shape characterization	37	5.2 %	197	3.6 %	2 360	6.8 %

Table 6.4: Comparison between the simulation and experimental data based on the mode shape of the TDEs. Difference calculated with LS Dyna[®] simulation as reference.

The points of interest are the displacements of the elements located at the welds and at the boundary face of the downstream flange. These results will be the foundation for submodeling of the downstream flange and welds. In the following chapter, the downstream flange is assessed using a

sub-modeling approach.

6.4 Conclusion

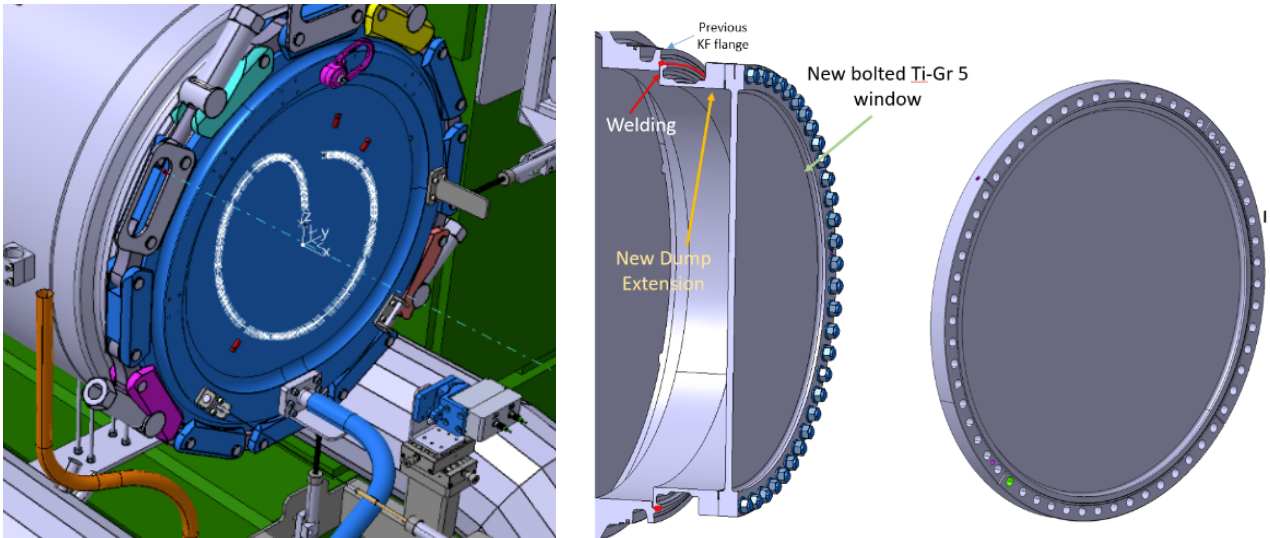
To summarize, the simulation of the Target Dump Externals upon beam impact creates the foundation for sub modeling of the downstream flange and welds. The result obtained is used to constrain the sub models representing the load condition of the components accurately.

The simulation of the Target Dump Externals is a multistep analysis using a weakly coupled thermo-mechanical explicit algorithm to simulate the shrink-fitting of the high-density cores segment in the vessel followed by the beam impact. The shrink-fitting was simulated using a diametral interference and the results showed good correspondence with calculations following the ANSI B4.1 standard and an implicit analysis in Ansys®. The simulation of the beam impact was performed by implementing FLUKA-energy density maps in the vessel, flanges and windows of the model. The energy deposition implementation was verified by comparing total energy and peak energy density, and showed precise interpolation between the energy density map and the model mesh.

Upon beam impact, the Target Dump Externals experience a dynamic response that can be decomposed as a superposition of mode shapes. The response results in a multiaxial stress state exposing the system to high-amplitude non-proportional stress oscillations. While the frequency of the mode shapes showed good correlation with experimental data, the amplitude of the oscillations was higher in the simulation, developing a conservative result when assessing the downstream flange and welds using a submodel approach.

Downstream Flange

In response to Nitrogen leaks found in the downstream window assembly, a new bolt-based mounting system was implemented during Long Shutdown 2 in 2020. To make this possible, a transition was welded onto the existing flanges, enabling the bolting feature. In figure 7.1a, the original clamping mechanism is seen and in figure 7.1b the flange extension and the new mounting mechanism is shown.



(a) Upstream flange and window showing the previous chain clamping mechanism that developed leaks. (b) Cross section of the downstream flange and window assembly showing the welded extension.

Figure 7.1: Downstream flange modifications during Long Shutdown 2

A new and simplified version of the downstream flange is proposed for implementation in the Run 3 operational dumps. Figure 7.2 shows this design, which allows for a direct transition from the vessel to the bolted window.

The primary points of interest for the flange assessment are the fillets in the thin area in the transition from the vessel towards the bolting area. These areas are marked with red circles in the figure on the right of 7.2. The reason for these areas to be specifically interesting is due to the stress concentrations

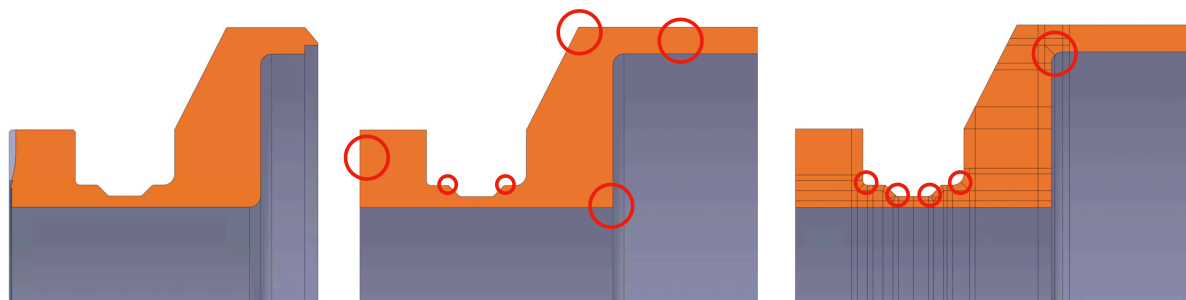


Figure 7.2: Cross-section view of the proposed design of the downstream flange. The figure on the left shows the unmodified geometry, the middle one shows the geometry after defeating. On the right, the geometry ready for meshing is visible.

that occur because of the transition between different thicknesses, crowding the stress flow lines in the component. Figure 7.4 describes the concept of flow lines and stress concentrations. Generally, abrupt transitions result in higher stress concentrations, which is important, especially for fatigue resistance.

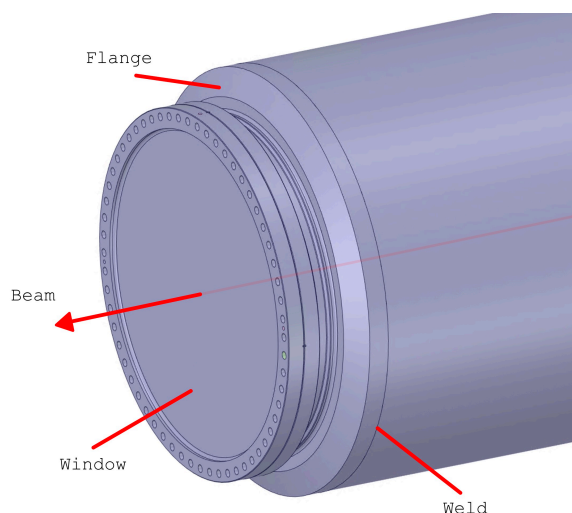


Figure 7.3: The proposed downstream flange design is welded to the vessel and the Titanium window is bolted to the flange face. Bolts are not shown.

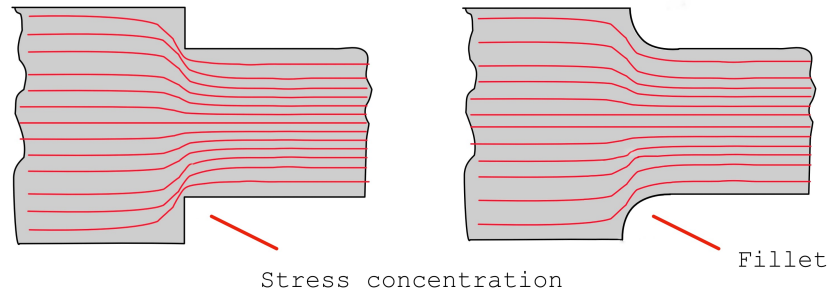


Figure 7.4: Illustration of stress flow lines in the geometry thickness transition. At abrupt transitions (left), the flow lines are crowded, resulting in higher stress concentrations. At smoother transitions (right), the flow lines are less crowded, resulting in lower stress concentrations

7.1 Simulation of the Downstream Flange

Recalling the workflow in figure 6.1 the result of the TDE simulation upon beam impact is used as a boundary condition for the simulation of the downstream flange. In the vessel of the TDE, a slice in the geometry records the nodal displacement of all nodes on the boundary face. Then, these nodal displacements are imported and applied to the face of the flange as boundary conditions. As a result, during the simulation the sub-model is loaded with the displacements that are generated in the vessel as a result of vibrations upon beam impact. Figure 7.5 illustrates the approach schematically.

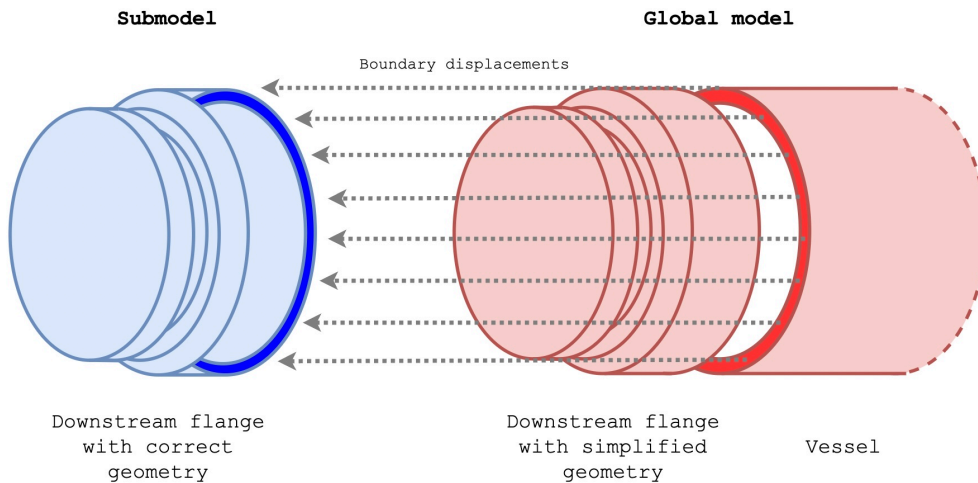


Figure 7.5: Illustration of the sub-modelling approach. The global model is shown on the right (full dump core). The sub-model is shown on the left in blue. The sub-model includes a more detailed geometry so that the stress in the fillets can be assessed.

The reason for using the sub-modeling approach is that only the downstream flange will be included in the model, which allows significantly finer mesh, properly capturing the stress gradients in the critical areas. This also allows the inclusion of geometry details such as fillets and grooves. For reference, the final mesh of the downstream flange contains a number of elements in the same range as for the global model of the TDE.

LS-Dyna[®] automatically performs interpolation in space and time, meaning that, despite the different element and time step sizes between the models, the displacements are imposed correctly on the submodel boundary. By default, the displacements are exported to a separate text file at a frequency of 1/10 of the output frequency. This means that the resolution of the displacement is a factor of 10 better than the output data that is available for post-processing. Figure 7.6 illustrates the concept schematically: the blue line represents the steps solved by LS-Dyna[®] that typically amount to hundreds of thousands over the simulation time. The red dots represent the data points that are written to the hard drive, which is the resolution available to the user for post-processing. This means that the user only sees a fraction of the data LS-Dyna[®] has solved and therefore needs to ensure that a proper output frequency is selected to capture the dynamic response correctly. The smaller orange dots represent the points at which the displacement data of the selected faces are written and applied as boundary conditions. In all the simulations that were performed, a comparison was carried out between nodes in the same location at the boundary face of the global model and the submodel. This comparison showed identical displacements, confirming a correct implementation of displacements in the submodel.

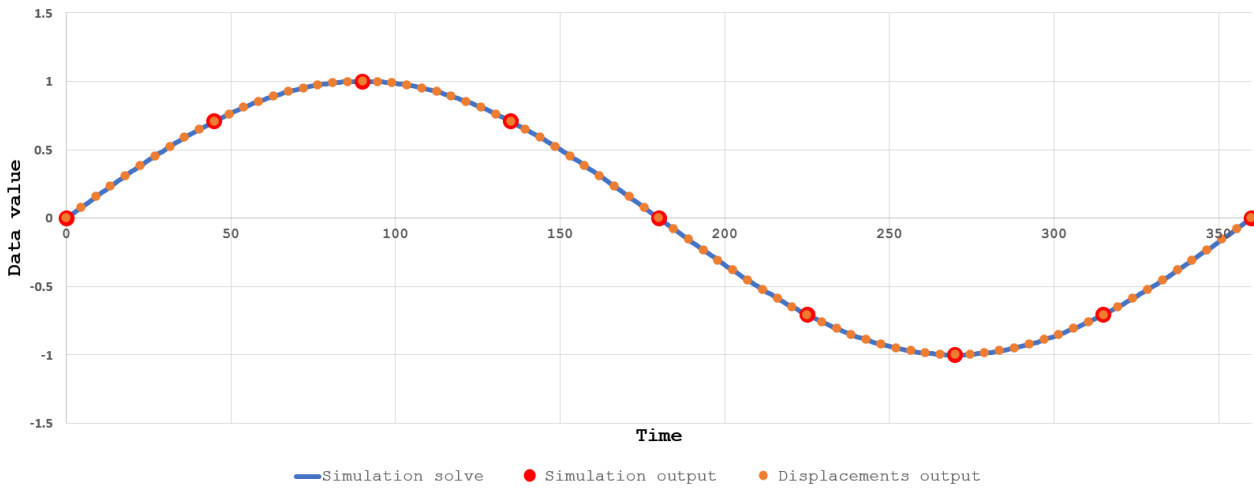


Figure 7.6: Illustration of different time-dependent triggers in explicit simulation. The blue line represents the steps where the model was solved (orders of hundreds of thousands), the small orange dots represent the export of displacements used for submodeling, and the red dots represent the steps where the simulation result was written to the hard drive.

In addition to the boundary displacements, the energy deposition as well as the internal pressure of 0.02 MPa is included in the submodel. Although the energy deposition represents less than 3 % of the total energy included in the global model (table 6.3) the energy deposition is expected to have a local effect. The downstream window is included in the submodel due to its inertia as well as its contribution to the structural stiffness of the assembly. It is implemented with a tied surface-to-surface contact to the flange.

The vast majority of the set-up time is used on the meshing of the flange. Meshing is an iterative process pursuing the best-performing mesh balancing the number of nodes and elements with the accuracy of the result. Especially for this detailed submodel, the element size is important, because it directly affects the time step size. Although the global model and the submodel have a similar number of elements, the amount of time consumed to solve the submodel will be substantially longer as a consequence of the reduced element size and thus the CFL condition to make the explicit solution stable [35].

CHAPTER 7. DOWNSTREAM FLANGE

To assist the meshing algorithm, the geometry is simplified (removal of exterior fillets and holes) according to the guidelines in A.3. A summary of the modifications can be found in A.5. Furthermore, the flange is divided into several bodies with simpler geometry to assist the meshing algorithm. This can be seen to the right in figure 7.2. Especially around the fillets and in the thin section of the flange (red circles in the right figure of 7.2) the mesh was given attention to properly capture stress gradients and peaks.

Since the beam impact simulation of the TDEs is the second step in a multistep simulation, the geometry at time 0 is slightly deformed due to the shrink-fitting of the high-density blocks. This means that the displacements transferred to the submodel are located on a face that is deformed, and thereby the contact algorithm responsible for identifying the boundary faces of the global model and submodel did not transfer the displacements of the misaligned faces correctly. Even though the change in nodal coordinates was in the order of hundredths of a mm it was enough for the algorithm to not capture the face and thereby transfer the displacements. This was solved by performing a 3D mesh projection of the deformed global model face onto the mesh of the flange perfectly aligning the faces of the models and thereby capturing the displacements correctly.

7.1.1 Results of Downstream Flange Simulation

A total of 18 simulations were performed on the three versions of flanges. This allowed the evaluation of the effects of different mesh settings and energy deposition applications. By comparing the results of the different flange designs, possible beneficial changes were mapped ensuring a comprehensive understanding of the behaviour of this part of the vessel.

The objective is to locate the critical areas where stress peaks occur. This is achieved by beginning with a coarse mesh that indicates the areas of interest. The global mesh settings are refined in a few iterations before locally refining the mesh to achieve peak stress convergence. A simplified overview of the mesh iterations can be seen in figure 7.7.

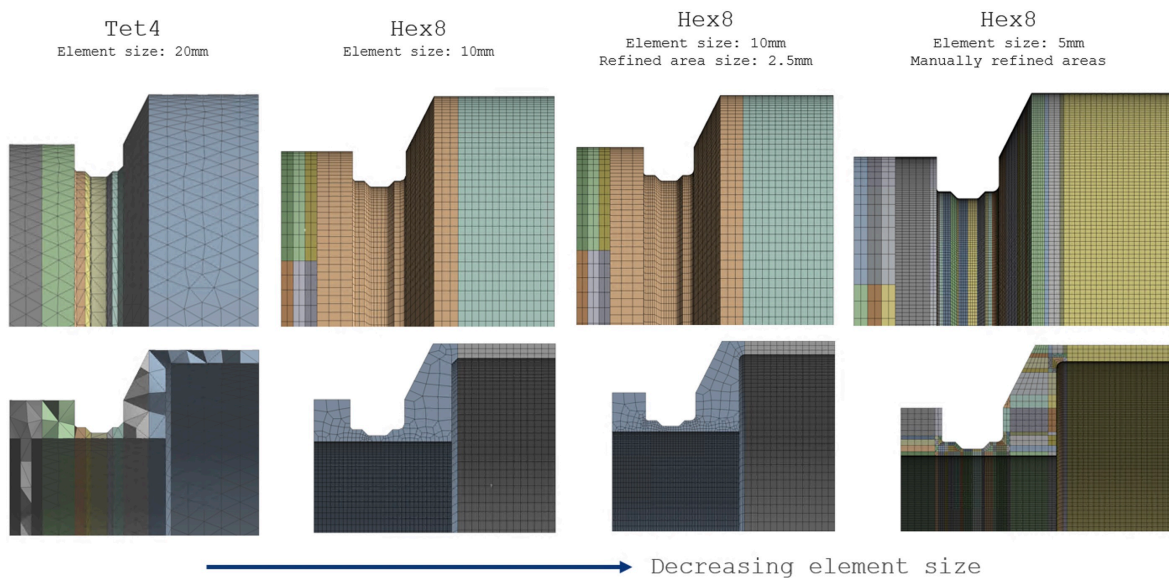


Figure 7.7: Mesh iterations of the downstream flange. From left to right in the process from localizing peak stress areas to peak stress convergence.

As anticipated, the fillets marked with red circles on the right in figure 7.2 experience the highest

CHAPTER 7. DOWNSTREAM FLANGE

levels of stress. This is especially the case for fillets in the transitional area of small thickness. Several mesh refinements were performed in these areas before achieving a stress peak convergence criterion that was defined as a maximum of 5% difference in the stress peak level per refinement.

The stress is driven primarily by the movement of the flange as a consequence of the vibration of the vessel pulling and pushing the flange boundary face. Due to the inertia of the flange and the window, the material experiences tension and compression as a function of time. An analysis was performed to assess the effect of energy deposition in the flange and showed that including the energy deposition in the simulation caused a 15% increase in stress peak. From a single element point of view, the load case of movement and energy in combination with geometry effects results in a complex stress state. However, the main contributor is the vibration resulting in a displacement in the longitudinal direction of the flange boundary face, consistently with the observations made in 6.3.1.

As seen in Figure 7.8 both the extended flange design implemented in the currently operational dumps (labelled "2018") and the first iteration of the new design (labelled with "2022-1") experience von Mises stress peaks close to the yield strength of the material at 450 MPa. The maximum peak stress in the "2018" geometry is located in areas with small thickness. The maximum von Mises stress is 420 MPa at the locations highlighted with red circles. Note that the stress peak changes location over time depending on the motion of the flange, meaning that other fillets are also exposed to high stress levels during the beam impact.

The first iteration for the new design shows similar patterns with stress peaks in the fillets, especially in the transitional area of small thickness. The stress peak is 2.7% lower at 409 MPa, indicating that this design faces stress conditions similar to the previous. This condition is somewhat expected given the many similarities in the two designs.

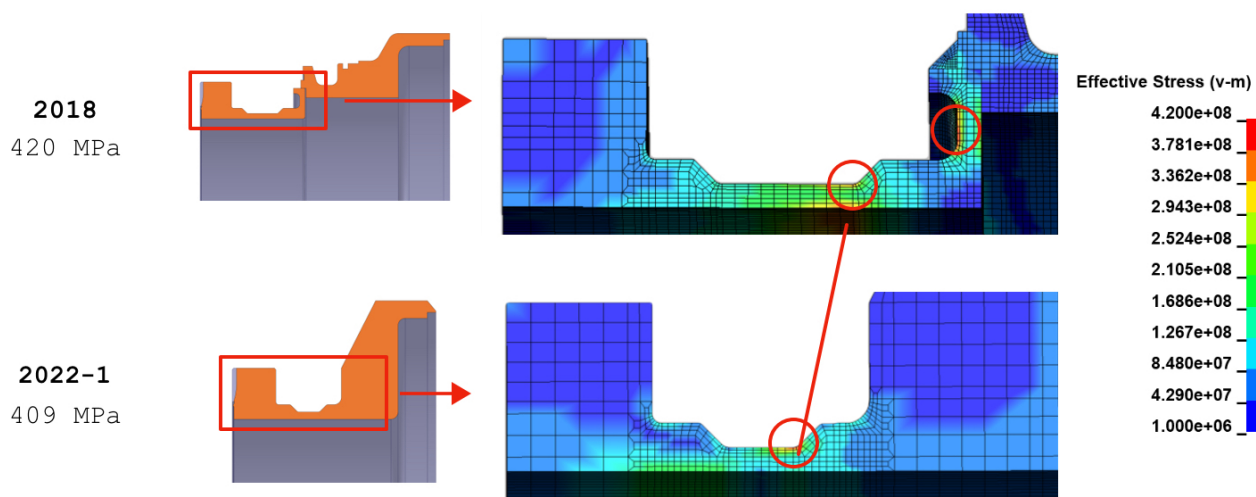


Figure 7.8: Time step at von Mises peak over 25 ms simulation time in 2022-1 and 2018 edition of flanges.

From a static point of view, both designs would only be exposed to stress levels in the elastic domain of the material. However, considering the many stress oscillations that occur as a result of a single beam impact multiplied by the number of impacts over the TDEs' lifetime, fatigue resistance becomes a concern.

CHAPTER 7. DOWNSTREAM FLANGE

Based on the concept of stress flow lines and the knowledge obtained from simulations regarding stress concentrations, a new geometry was created focusing on reducing stress peaks. Conceptually, the update followed the first iteration of the new design, with modifications in critical areas highlighted by the previous simulation. The update included a design where the transition area was increased to a uniform thickness with an increase in adjacent fillet radius. The two designs are shown in figure 7.9.

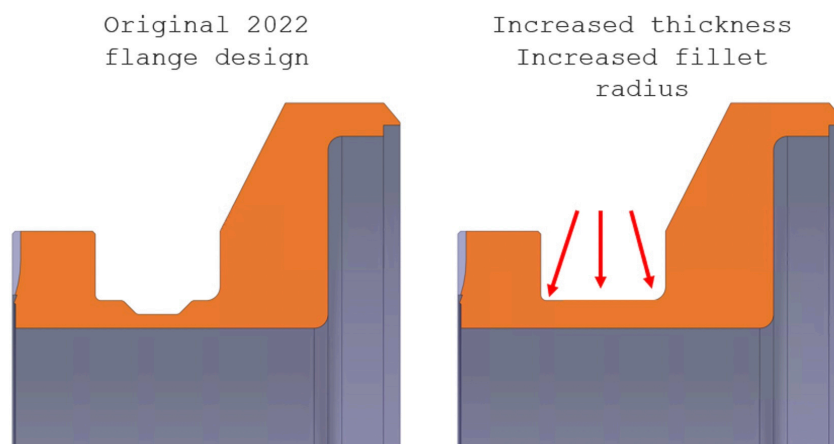


Figure 7.9: Flange modifications to reduce stress peaks and minimize stress concentrations.

An evaluation of the updated design was performed following a similar iterative process, highlighting the location and level of stress peaks after convergence. The increase in thickness and removal of fillets in the transition area resulted in a 42% reduction in the stress peak. Furthermore, the update changed the location of the stress peak. As seen in Figure 7.10 the stress peak is reduced from 409 MPa to 236 MPa. The stress peak area has changed to a fillet with a larger radius and material thickness, for even better resistance to crack initiation and propagation.

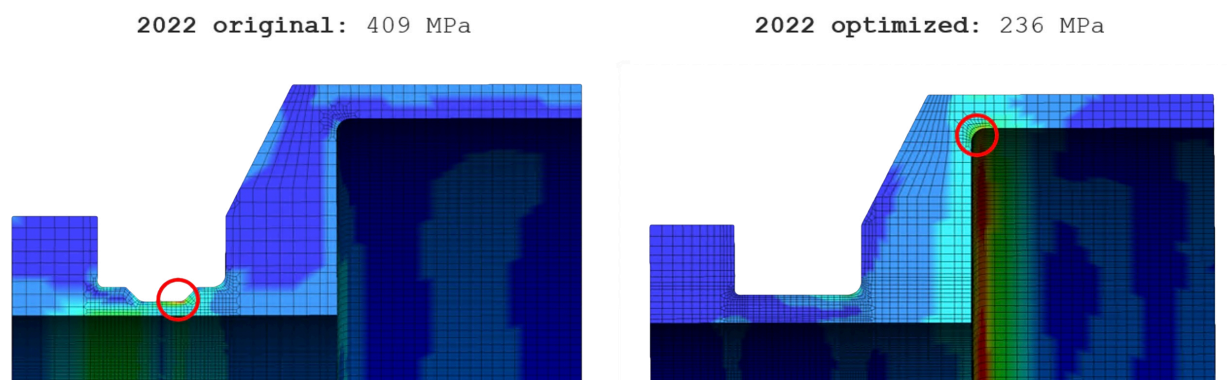


Figure 7.10: Comparison of the von Mises stress peak and location between the original 2022 design and the updated 2022 design with uniform thickness and increased radii.

This updated design has a safety factor of more than 2 compared to the yield strength of 450 MPa of 318LN (4.1). Therefore, the updated design with uniform thickness and an increase in fillet radius was adopted as the new baseline design. The following chapter will show the procedure to assess its fatigue resistance.

7.2 Damping of Simulation Data

To properly assess the fatigue resistance of the flange, a correct damping of the dynamic response must be applied. Excluding the effect of damping would substantially increase the number of high-stress amplitude oscillations, producing an overly conservative assessment. The damping ratio of the system can be extracted by processing the data from the many different sensors that are installed on the operational dumps. This parameter can then be applied to the results of the simulations during post-processing for a more realistic interpretation of the evolution of the stress oscillation after beam impact.

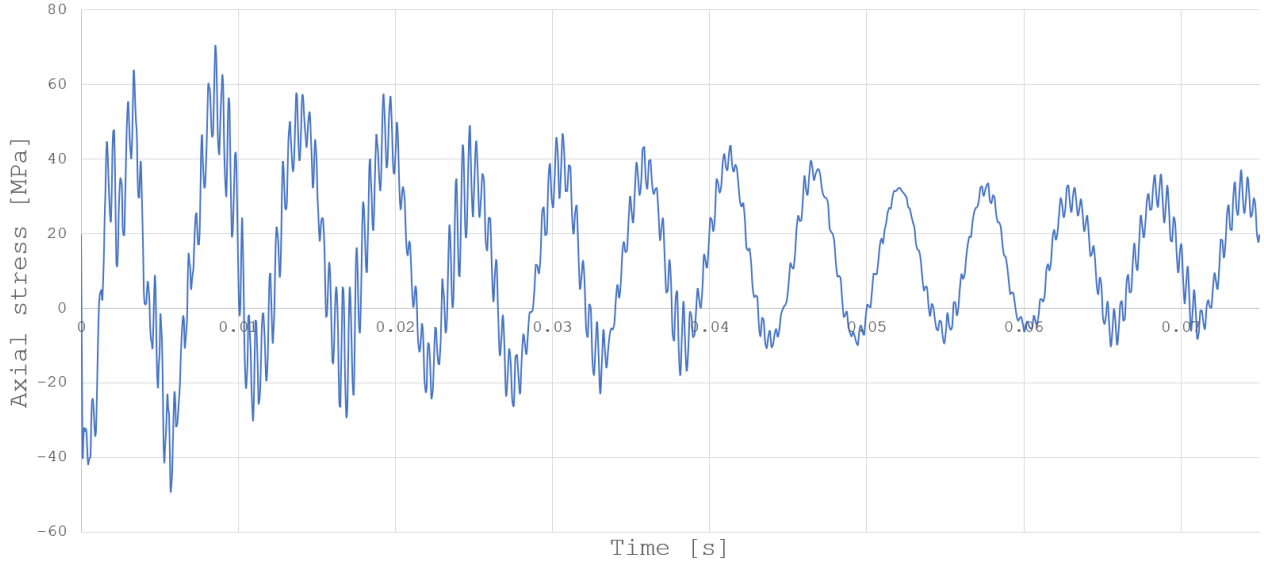


Figure 7.11: Axial stress in an element that is located close to the longitudinal center of the dump vessel.

Figure 7.11 shows the axial stress in an element that is located close to the longitudinal center of the dump vessel. In section 6.3.1 it was concluded that the longitudinal expansion drove the stress peaks in the dump. It should be noted that the amplitude of the axial stress oscillations is constant after 0.5 s, not representing the damping of the system correctly. This is due to the unconstrained setup of the TDE simulation upon beam impact and to the absence of a damping factor implementation in the simulation.

Data from the sensors that are installed on the operational dumps can be used to determine this parameter. Figure 7.12 shows the velocity of the dump upon beam impact measured with a 10 kHz Laser Doppler Vibrometer (LDV) pointing at the upstream window. It is seen that the velocity data is oscillating around 0 m/s and is decaying exponentially. This behaviour corresponds to an underdamped oscillation pattern with a damping ratio (ζ) less than 1.

The damping ratio can be estimated by post-processing this data in Python. This process is explained in further detail in (A.9). The data above 0.3 s is filtered out since the peaks of the oscillations at low amplitude become unclear. By identifying the peaks of the velocity oscillations with logic in the script, an exponentially decaying function can be fitted to the peak data points. As seen in figure 7.13b, the general expression describing the exponential decay of an underdamped system is described as follows:

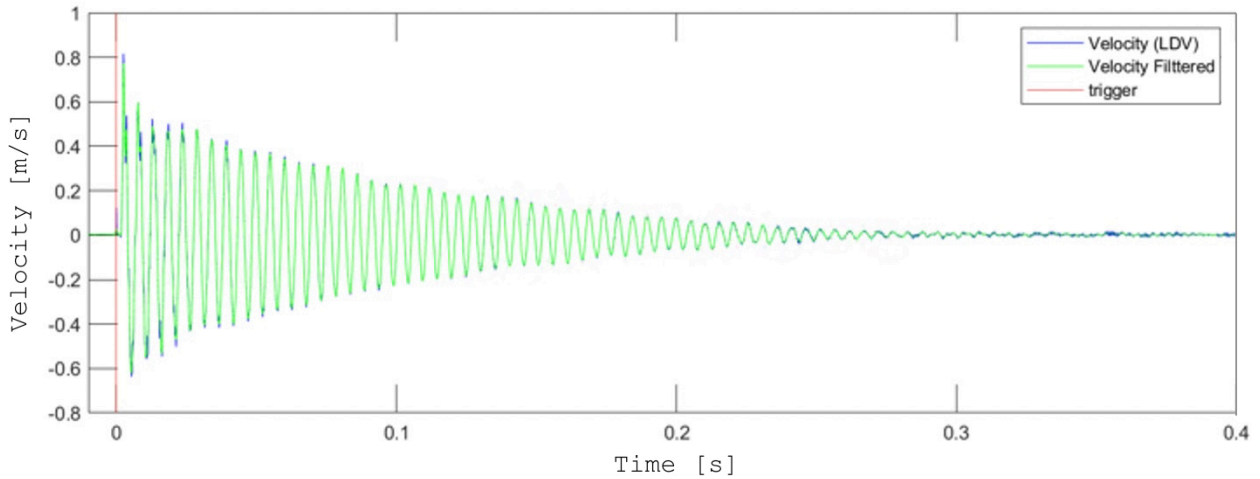


Figure 7.12: Target Dump External Laser Doppler Velocimeter measurements of dump velocity pointing at the upstream face of the dump. Data from the highest energy dump so far during Run 3 the 14. november at 386 MJ.

$$f(t) = C \cdot e^{-\zeta\omega_n t} \quad (7.1)$$

Where C is a constant, ζ is the damping ratio, and ω_n is the natural frequency of the system.

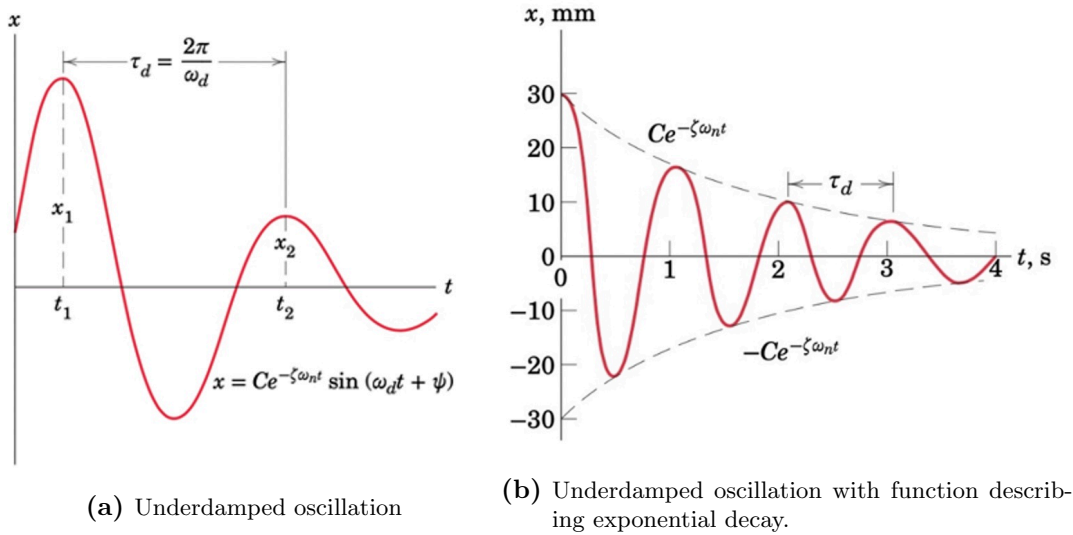


Figure 7.13: Damping theory describing underdamped oscillations. Figure 7.13a shows expression for underdamped envelope. Figure 7.13b shows expression for exponential decay in damped systems. [17].

By fitting the peaks of the velocity data to an exponentially decaying function, the C constant as well as a product of ζ and ω_n are obtained. To ensure that these values reliably represent the damping, the first oscillation of the data is filtered out due to a high peak that would increase the decay of the system, making it less conservative. In figure 7.15 the peaks (red dots) used for fitting the

CHAPTER 7. DOWNSTREAM FLANGE

exponentially decaying function (orange) is shown. The expression for the curve fitted to the peak data points is as follows:

$$f(t) = 0.63 \cdot e^{-10.19 \cdot t} \quad (7.2)$$

Since -10.19 is a product of ζ and ω_n the expression for ω_n along with a Fast Fourier Transformation (FFT) of the data to define ω_d is obtained to solve two equations with two unknowns. Figure 7.14 shows a clear spike at 190.6 Hz corresponding to the longitudinal frequency of the TDEs 6.4. 190.6 Hz is the damped frequency of the system ω_d . The two equations with two unknowns ζ and ω_n are:

$$f(t) = C \cdot e^{-\zeta \omega_n t} \quad (7.3)$$

$$\omega_n = \frac{\omega_d}{\sqrt{1 - \zeta^2}} \quad (7.4)$$

Using boundary conditions in equation 7.2 ω_n can be expressed by ζ in 7.3 and substituted into equation 7.4:

$$f(0.05) = 0.63 \cdot e^{-10.19 \cdot 0.05} = 0.378 \quad (7.5)$$

$$0.378 = 0.63 \cdot e^{\zeta \omega_n \cdot 0.05} = \frac{10.21}{\zeta} \quad (7.6)$$

$$\frac{10.21}{\zeta} = \frac{190.6 \cdot 2\pi}{\sqrt{1 - \zeta^2}} \quad (7.7)$$

By solving the equations, the natural frequency ω_n is found to be 190.9 Hz and the damping ratio ζ is found to be 0.0085, which corresponds to a damping of 5.3 %. The amplitudes of the oscillations decrease by 5.3 % between each successive peak. This is applied to the simulation results in the fatigue assessment to properly account for damping in the following chapter.

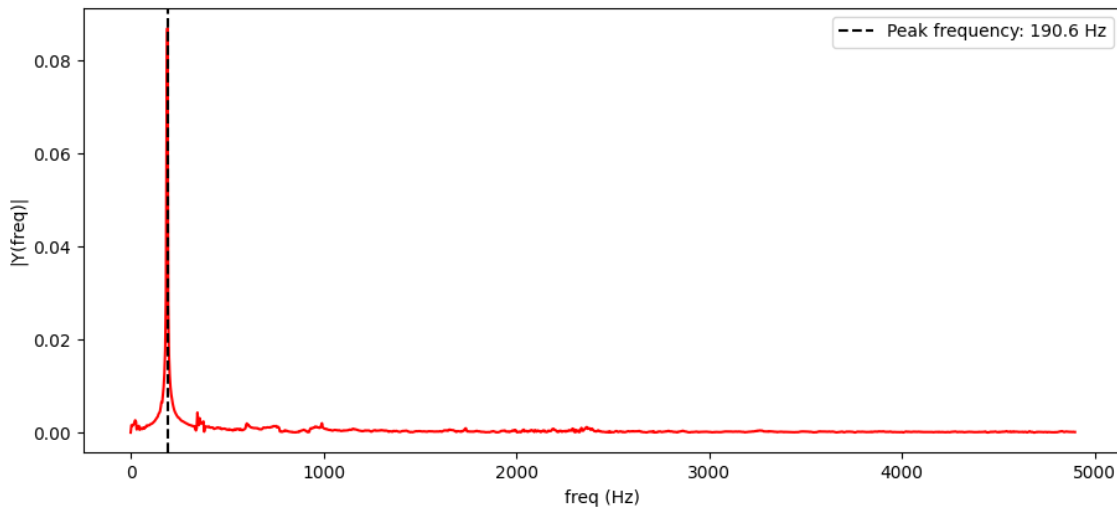


Figure 7.14: Fast Fourier Transformation of velocity data from figure 7.12 to obtain ω_d .

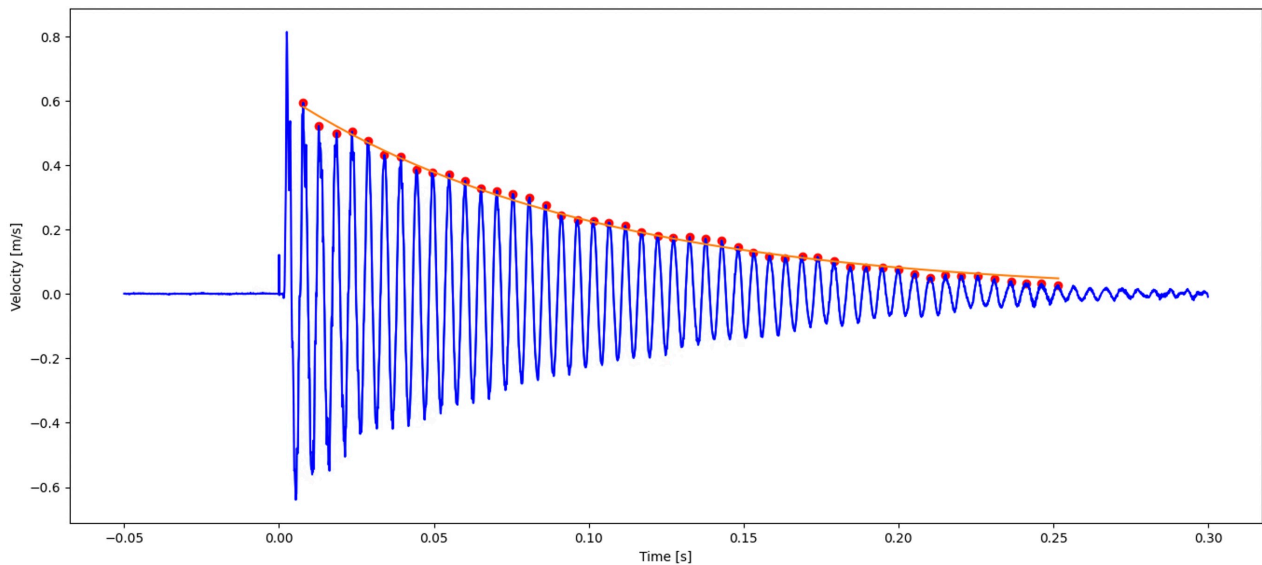


Figure 7.15: Post processing of velocity data. The blue line shows data, red dots represent peaks used for fitting the exponentially decaying curve in orange.

7.3 Multi Axial Fatigue

As a consequence of the downstream flange being exposed to varying loads over its lifetime, fatigue resistance is an important aspect. Fatigue failure can be divided into three stages illustrated in figure 7.16. The first stage is the formation of cracks, which is driven by cyclic plasticity in the local area. The crack formation depends on many factors, including type of loading, surface roughness, etc. The second stage is the propagation of the crack, where the crack due to the reduced area of the forces to act upon experiences recurrent plastic deformation resulting in the crack growing. The last stage is fracture, where the crack has grown to a critical size, resulting in the component no longer being capable of withstanding the loads.

Typically, fatigue in the elastic domain is assessed using S-N curves that are based on experimental data where test specimens are exposed to uniaxial stress oscillations at different stress ranges.

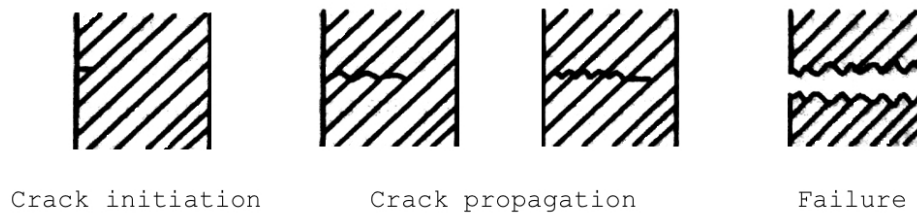


Figure 7.16: Fatigue stages showing crack initiation, crack propagation to a critical size before fracture.

However, this is not directly applicable to the fatigue assessment of the downstream flange due to several reasons. Despite the stress levels being driven by the longitudinal expansion of the vessel, the local stress tensor is a product of the loading of the downstream flange and the way the stresses are developing through the geometry. This means that the principal direction of stress in an element varies as a function of time and position. Due to the many vibration modes transferred to the flange, the stress state is multiaxial and nonproportional, since the frequency of the modes is out of phase and the principal direction of stress varies.

To cope with the complicated fatigue aspect, nCode® DesignLife (nCode®) is incorporated into the simulation workflow. nCode® is capable of reading the output simulation data from LS-Dyna® and post-process it in order to yield fatigue life predictions. nCode® uses an element-based fatigue calculation engine, which means that a life prediction will be estimated for each element of the model.

Since stress concentrations and crack initiation start from the surface, only surface elements are used in the assessment. nCode® automatically detects the surface elements of the model and rotates the stress tensors of these elements so that one axis is normal to the surface of the element. This simplifies the 3D tensor into a 2D tensor that is computationally lighter to process. nCode® determines the principal direction of stress in the element and uses the rainflow counting method along with Miner's rule to convert the simulation data into fatigue life predictions. Ideally, a multiaxial plasticity model accounting for hardening should be applied, but since the maximum von Mises stress in the simulation is well below the yield strength a more complicated model for the fatigue life was not adopted.

nCode® is fed with the extrapolated and damped simulation results of the downstream flange. The concept is illustrated in figure 7.17. The simulation data (blue) shows signs of constant amplitude towards the simulation ending not representing the actual damping of the system. To more accurately assess the fatigue aspect, the first two oscillations are included in nCode® without damping. The data corresponding to the second oscillation is duplicated and damped progressively by 5.3 % for every subsequent oscillation. This is better illustrate in figure 7.17: the orange box covers the simulation data that is selected, duplicated progressively damped. 24 oscillations are considered for the fatigue analysis corresponding to a stress peak reduction of 66 %.

The manipulated data are arranged in a duty cycle configuration. The duty cycle option allows one to replicate the loading history. This is done 1600 times, corresponding to 400 dumps per year over 4 years, which is the required lifetime of the downstream flange.

Four different S-N curves obtained from a previous study on the vessel were used to compute the predicted lifetime [12], [29], [6], [21]. The S-N curves are shown in figure 7.18.

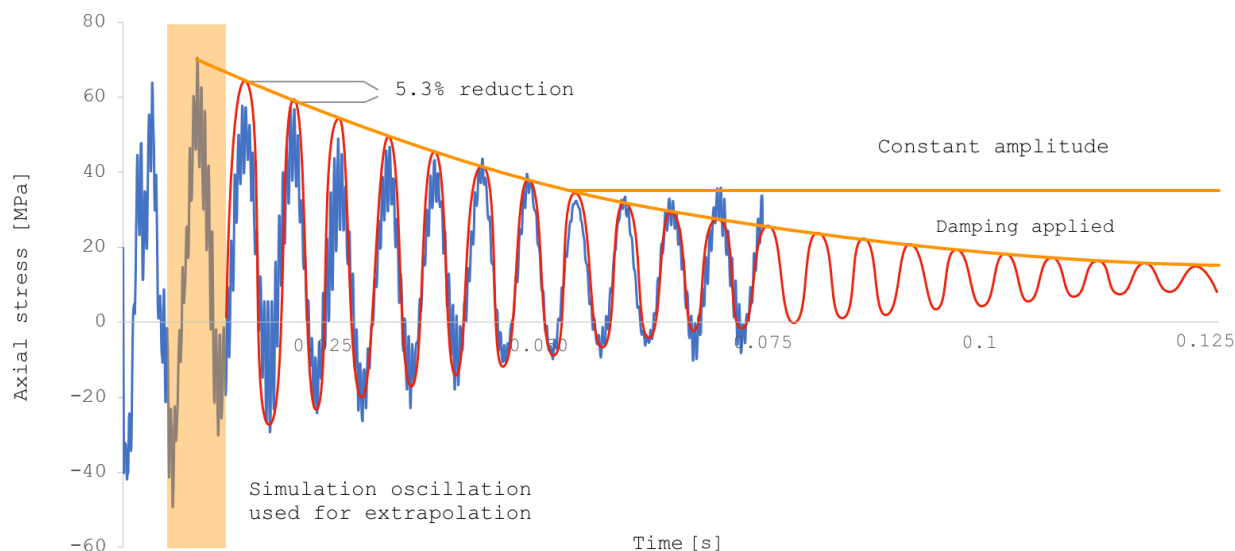


Figure 7.17: Conceptual description of damped extrapolation of simulation data. The simulation data from the second oscillation (covered by orange box) is duplicated and progressively damped by 5.3 % for every subsequent oscillation.

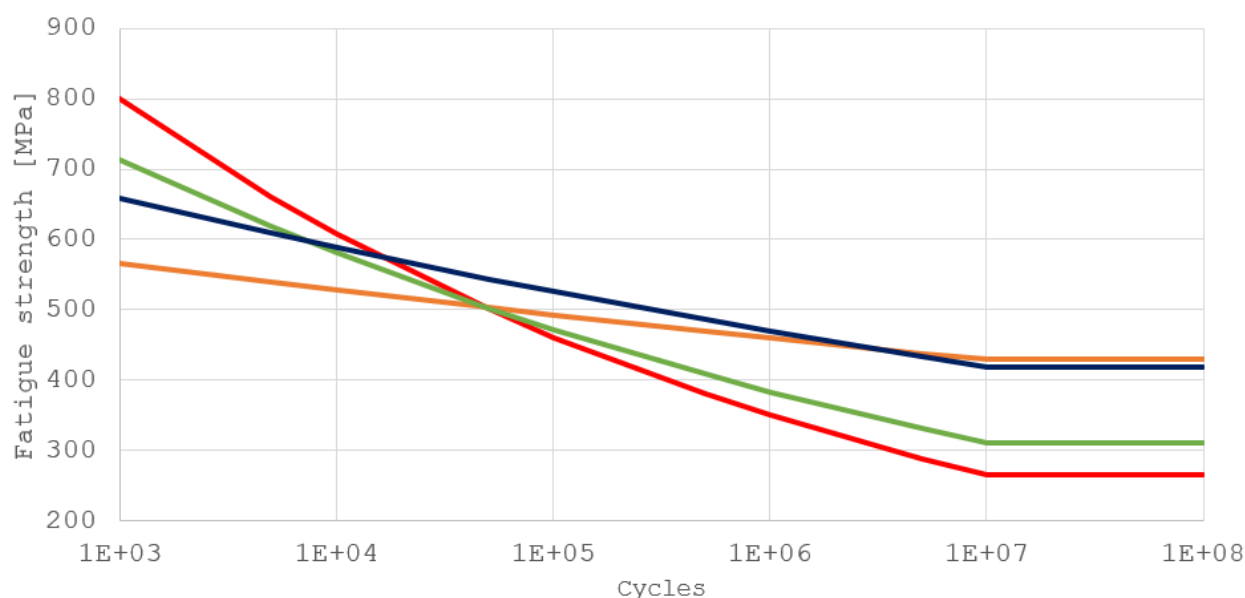


Figure 7.18: 318LN S-N curves used for the fatigue assessment of the downstream flange [6], [12], [29], [12].

All S-N curves adopted for the fatigue assessment have endurance limits at 10 million cycles. Looking at figure 7.18 the S-N curve for bending (red) has the lowest endurance limit at 265 MPa which means that below this stress level, the flange is estimated to have infinite life. However, the other S-N curves are more conservative in the low-cycle fatigue domain of fewer than 50 000 cycles. The manipulated simulation data fed to nCode® correspond to 38 400 longitudinal cycles taking into account 24 oscillations per dump and 1600 dumps.

7.3.1 Fatigue Life Results

nCode[®] yields an element-based fatigue life prediction using the manipulated simulation data obtained in the LS-Dyna[®] simulation of the downstream flange upon beam impact. The life prediction of the component is illustrated in 3D enabling a detailed evaluation of the fatigue results.

Looking at fatigue results for the original design of the downstream flange for the Run 3 spares in figure 7.19, it is clearly highlighted that the presence of a low radius fillet in the transition area reduces the lifetime of the component substantially. The result is based on the S-N curve that yields the most conservative life prediction. The blue colour represents the elements where infinite life is predicted due to stress oscillation levels being lower than the endurance limit. It can be observed that the vast majority of the component has infinite life.

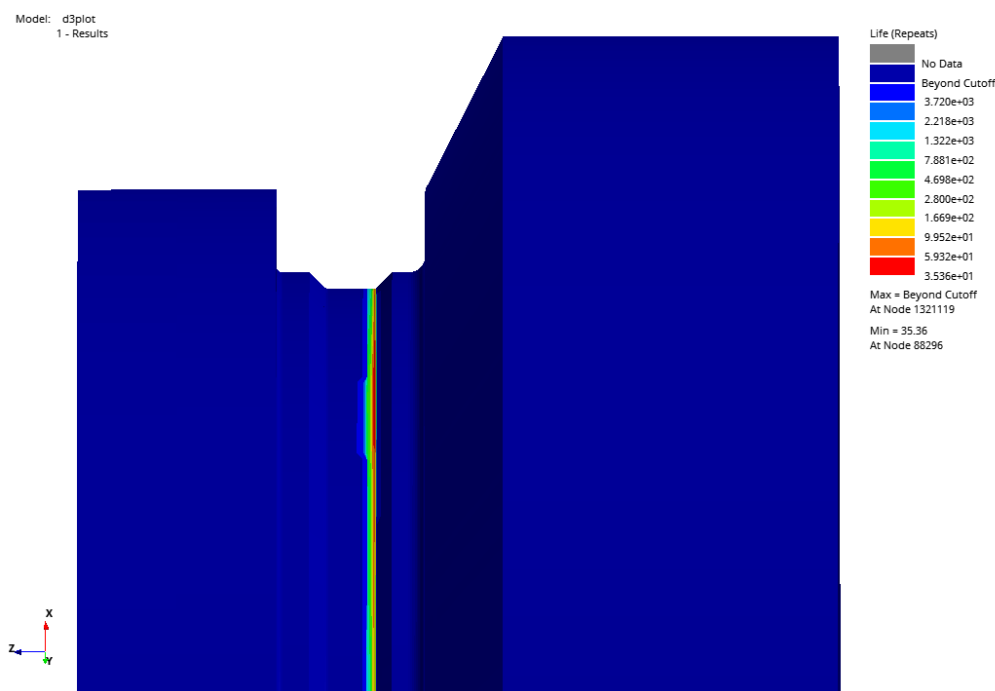


Figure 7.19: Fatigue life prediction for original flange design.

In the critical area of the fillet, a life prediction of 35 cycles was estimated. It should be noted that 35 cycles refer to 35 times the loading history fed to the calculation engine, meaning that the downstream flange even in the original configuration would have a predicted life 35 times as long as the required lifetime (400 dumps per year over 4 years).

However, looking at figure 7.20 of the optimised design of the downstream flange design with uniform thickness in the transition area and increased fillet radii, it is seen that infinite lifetime of all elements in the model is predicted. Recalling from 7.1.1, a 42% reduction in the von Mises stress peak from 409 MPa to 236 MPa between the original and updated design drastically extends the predicted fatigue life of the downstream flange.

While the fatigue assessment of the original downstream flange design showed a minimum safety factor of 35 the updated design has infinite life. The fatigue aspect was non-critical for both configurations but the updated design has indisputably a better performing design reducing stress peak levels and thereby extending its life.

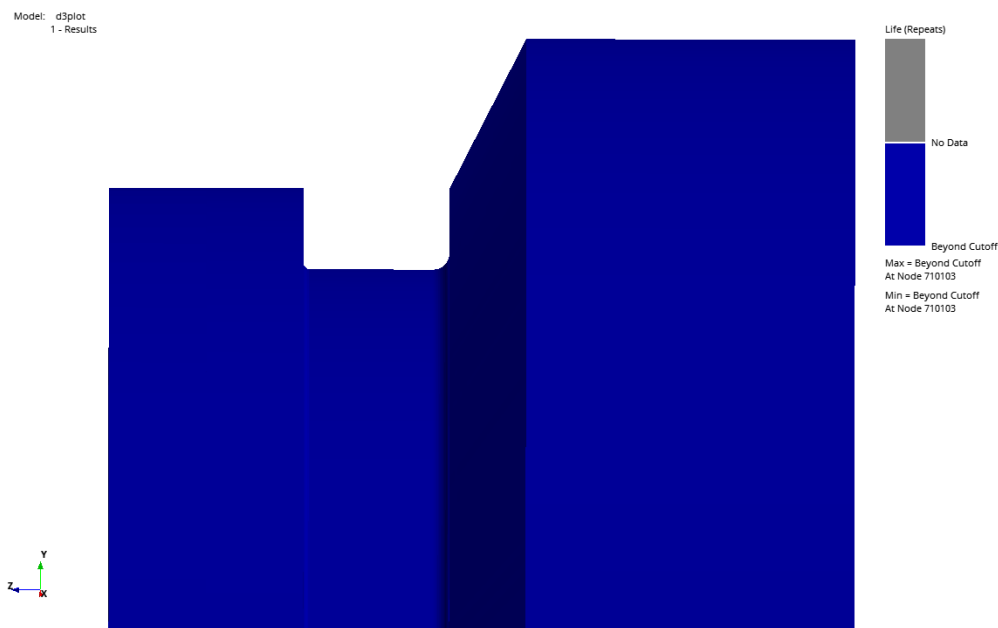


Figure 7.20: Fatigue life prediction for original flange design.

7.4 Conclusion

A comprehensive study of the response of the TDEs upon beam impact created the foundation for assessing the proposed downstream flange design for the manufacturing of the Run 3 spares. Using a submodeling approach to allow detailed meshing of the component an understanding of the behaviour of the downstream flange upon beam impact was developed, highlighting critical areas of the design. A study of several designs for the downstream flange revealed that the originally proposed flange design for the Run 3 spares had similar performance as previous generations, with areas exposed to von Mises stress levels close to the yield strength of the material. Based on the results obtained during the assessment, an updated design was suggested, with a uniform flange thickness in the transition area and an increase in fillet radii. The updated design resulted in a 42 % reduction in the von Mises stress peak increasing the safety margin with respect to the yield strength from 1.1 to 1.9. With the use of Laser Doppler Vibrometer data obtained from the currently installed dump, a damping ratio of the system was found, allowing a multiaxial fatigue assessment to be performed. Both the originally proposed and updated design is compliant with the fatigue life requirement of 1600 full-intensity beam dumps. However, the updated design has infinite life in all areas, whereas the originally proposed design has a fatigue life prediction of 35 times the requirement.

The updated design is verified for both static and fatigue and is implemented as the new baseline of the downstream flange design for the manufacturing of the Run 3 spare dumps.

Circumferential Welds

The proposed vessel configuration for manufacturing of the Run 3 spares consists of 11 segments and the upstream and downstream flanges. The segments are joined together by 12 circumferential welds performed after shrink fitting the high-density blocks inside them. The welds are partial penetration butt welds (also referred to as *socket welds* in the literature) with a penetration depth of 6.65 mm. This is slightly more than half of the thickness of the vessel of 12 mm. The partial penetration configuration shown in figure 8.1 is chosen for several reasons:

- Handling and alignment. After the segments are shrink-fitted with the high-density graphite blocks, they weigh close to 500 kg each. They are welded in two paths using totally mechanized Gas Metal Arc Welding (GMAW) with active gas, and to ensure proper alignment, the male-female configuration of the parent parts is adopted.
- Protection of the graphite blocks during the welding process. The chosen geometry protects the graphite blocks from being exposed to the welding process, which could potentially damage them.
- Inaccessibility after welding. Considering a full penetration weld, the inside of the vessel would be inaccessible after welding due to the high-density blocks, preventing inspection or machining of the weld from the inside.

The partial penetration weld introduces several challenges, but stress concentration due to the abrupt thickness transition carrying the loads is the main aspect of concern. A structural failure in the welds would be critical, and in the worst-case scenario lead to a shutdown of the LHC.

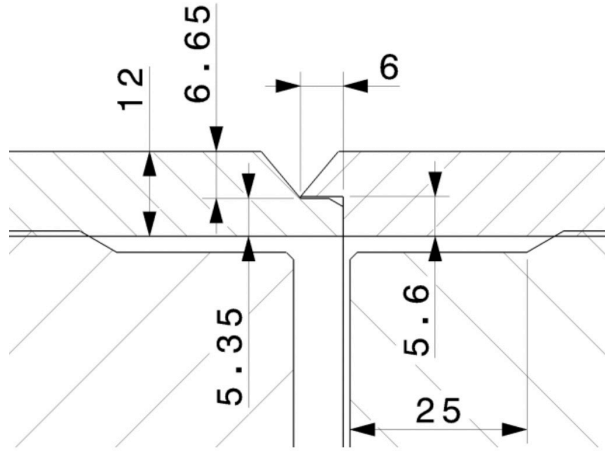


Figure 8.1: Cross section showing the current weld configuration with a weld depth of 6.65 mm. The thickness of the vessel is 12 mm. The shrink-fitted core elements are shown in the figure

8.1 Eurocode 3 Static Verification

The Eurocode 3 [5] standard contains a dedicated section for the design of joints in steel structures focusing on a wide variety of weld geometries and load cases that yield different methods and verification standards. The standard distinguishes between static verification and fatigue verification. Before fatigue assessment, a static verification is performed using the location and time step of the welds that are subjected to the highest stress levels according to the standard.

The weld type in Eurocode 3 that most closely describes the welds in the TDEs is a fillet weld exposed to normal and shear stress in the planes that are perpendicular and parallel to the weld. However, this type of weld does not capture the exact behaviour of the weld in the TDEs, as the partial penetration introduces stress concentration. Despite the fact that the method is not fully representative, the load cases are similar. To gain a sense of the margin with respect to the Eurocode 3 standard, the verification is performed using the results from the simulation of the TDE in 6.3.1 along with several assumptions described throughout the chapter. A similar assessment was previously adopted for weld verification of the downstream flange extension in 2018 [24].

In the static verification process obtained from Eurocode 3, there are two separate design resistance criteria. These consider different components of the stress tensor in the weld. In this method, the forces transmitted by a unit length of weld are resolved into components parallel and transverse to the longitudinal axis of the weld and normal and transverse to the plane of its throat. The orientation of the stresses defined in Eurocode 3 can be seen in figure 8.2.

The design resistance of the fillet weld will be sufficient if both following conditions are satisfied:

$$\text{Eurocode criterion I:} \quad [\sigma_{\perp}^2 + 3 \cdot (\tau_{\perp}^2 + \tau_{\parallel}^2)]^{0.5} \leq f_u / \beta_w \cdot \gamma_{M2} \quad (8.1)$$

$$\text{Eurocode criterion II:} \quad \sigma_{\perp} \leq 0.9 \cdot f_u / \gamma_{M2} \quad (8.2)$$

Where:

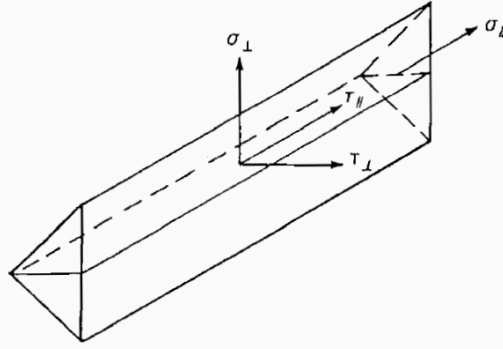


Figure 8.2: Stresses on the throat section of a fillet weld [5].

- $\sigma_{\perp}$ is the normal stress perpendicular to the throat.
- $\sigma_{\parallel}$ is the normal stress parallel to the axis of the weld.
- $\tau_{\perp}$ is the shear stress (in the plane of the throat) perpendicular to the axis of the weld.
- $\tau_{\parallel}$ is the shear stress (in the plane of the throat) parallel to the axis of the weld.
- f_u is the nominal ultimate tensile strength of the weaker part joined.
- β_w is the appropriate correlation factor.
- γ_{M2} is a partial safety factor.
- f_u is 660 MPa [21].
- β_w is set to 1 (A.6).
- γ_{M2} is set to 1.25 (A.6).

This yields design resistance criteria as follows:

$$\text{Eurocode criterion I:} \quad [\sigma_{\perp}^2 + 3 \cdot (\tau_{\perp}^2 + \tau_{\parallel}^2)]^{0.5} \leq 528 \text{ MPa} \quad (8.3)$$

$$\text{Eurocode criterion II:} \quad \sigma_{\perp} \leq 475 \text{ MPa} \quad (8.4)$$

If the stress levels in the welds are below the design resistance, the welds will be statically verified. To feed the design criteria with the correct stress tensor values at the weld locations, the simulation results of the TDE response upon beam impact obtained from section 6.3.1 have to be post-processed before being fed into the criteria.

8.2 Post Processing of Simulation Results

Using a Python script, the elements of the model located in the weld locations ± 3 cm on each side belonging to the vessel body are extracted for the assessment. The selection of elements for the assessment can be seen in figure 8.3. Since the vessel in the TDE simulation is modelled with a

CHAPTER 8. CIRCUMFERENTIAL WELDS

uniform thickness at the weld locations, using the stress tensors obtained from the result is based on the assumption that the stress levels in the welds are not affected by stress concentration factors. As a consequence, the stress levels are simply scaled by the area reduction that the forces act on, that is, the ratio of the thickness reduction:

$$12 \text{ mm} / 6.65 \text{ mm} = 1.8 \quad (8.5)$$

While this is clearly an excessive simplification, this preliminary assessment is instrumental in providing an idea of the most critical weld locations in the vessel.

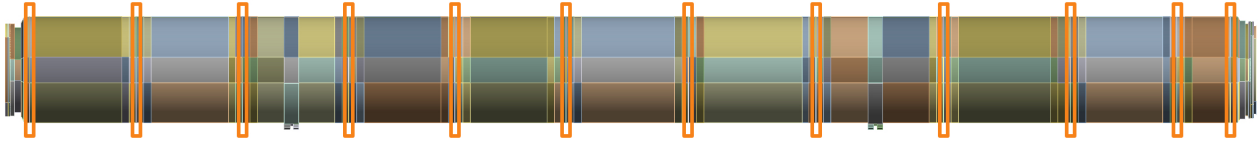


Figure 8.3: Element selection for Eurocode 3 assessment. Elements inside orange boxes correspond to the weld locations and are used for assessment.

As seen in 8.2, the Eurocode 3 standard refers to a coordinate system that is tangentially aligned to the weld. For the case of the TDE, the welds are circumferential, meaning that to perform the weld assessment, the stress tensors must be rotated in order to align with a local tangential coordinate system instead of the original global coordinate system in the simulation. Therefore, as conceptually visualized in figure 8.4, the stress tensors of the model are rotated around the longitudinal axis of the TDE from a Cartesian coordinate system to a cylindrical coordinate system based on the angle of the element in the vessel.

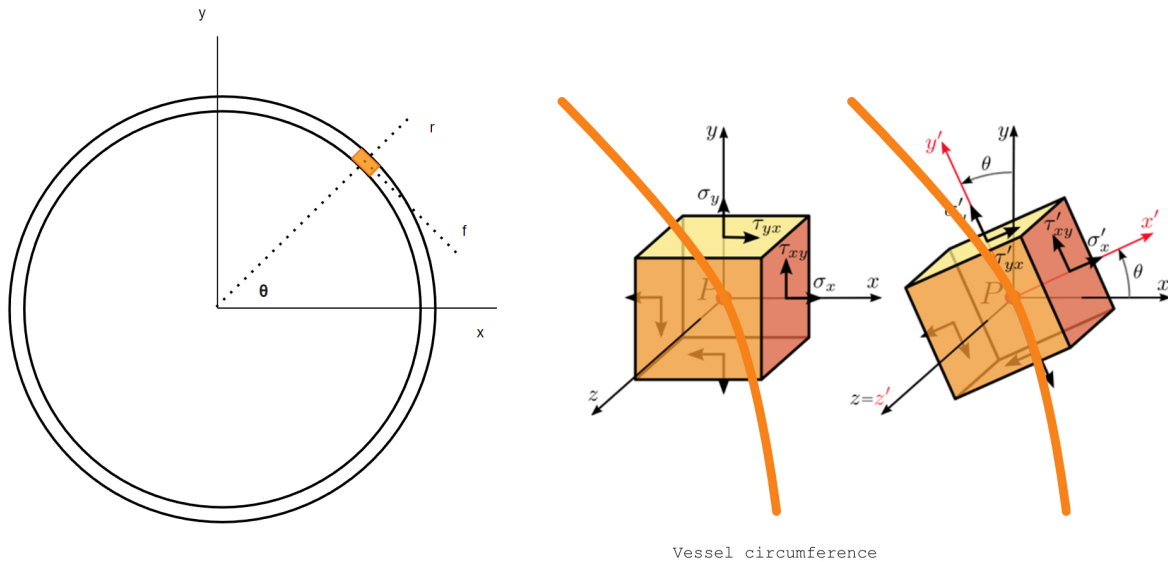


Figure 8.4: Illustration of stress tensor rotation around longitudinal axis. Left: cross-section view of the vessel. Angle is determined based on x - and y coordinates of the element. Right: Based on the angle of the element the stress tensor is rotated into a cylindrical coordinate system compliant with the Eurocode 3 application.

CHAPTER 8. CIRCUMFERENTIAL WELDS

The angle is determined on the basis of the x - and y coordinates of the element in the model. This can be done since the axial direction is aligned with the cylindrical center of the vessel. The rotations of the stress tensors are performed after the filtering of the elements from the TDE simulation to minimize the computational requirements for this operation. The stress tensors are rotated using a rotation matrix, and the rotation can be described as follows:

$$\sigma' = Q^T \cdot \sigma \cdot Q \quad (8.6)$$

Where:

- σ' is the rotated stress tensor.
- Q is the rotation matrix and Q^T is the transposed rotation matrix.
- σ is the original stress tensor (Cartesian).

The rotation matrix to rotate the stress tensor around the z-axis (axial axis of the TDE) is as follows:

$$Q = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (8.7)$$

Inserting Q into equation 8.6 the complete matrix operation is obtained:

$$\begin{bmatrix} \sigma_{rr} & \sigma_{rf} & \sigma_{rz} \\ \sigma_{fr} & \sigma_{ff} & \sigma_{fz} \\ \sigma_{zr} & \sigma_{zr} & \sigma_{zz} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}^T \cdot \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix} \cdot \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (8.8)$$

The operation is performed on all elements in the 12 weld locations, resulting in close to 20 million stress tensor rotations distributed over 502 time steps. After the stress tensor rotation, the script subtracts the pre-stress from the results. This is because the welds are performed after the shrink fitting and therefore they are only subject to the loads upon beam impact. The script automatically feeds the Eurocode 3 design resistance criteria (equation 8.1, 8.2) with the correct components of the stress tensors, resulting in a static assessment for each element. The components fed to the script are seen in figure 8.5.

As previously stated, the stress component product obtained by 8.1 and 8.2 is scaled by the area reduction factor (1.8), to closer represent the stress level in the weld due to the partial penetration. With built-in logic, the maximum value of the two Eurocode 3 criteria for every weld is extracted. In this way, the design resistance of the welds is based on not only the most critical weld but also the most critical circumferential position at the most critical time step. The precision is limited to the mesh resolution, but with 66 elements along the circumference of the vessel, the peak is expected to be captured accurately.

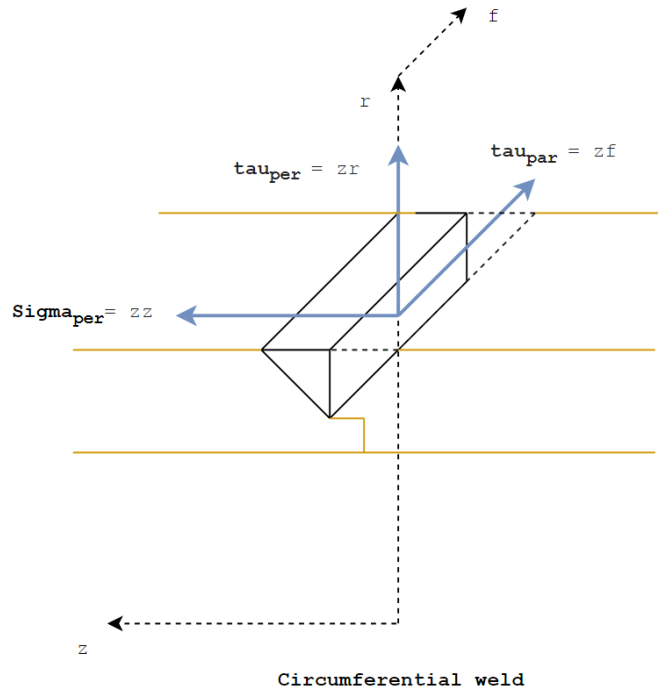


Figure 8.5: Stress tensor components processed by the script in compliance with the orientation prescribed by the Eurocode 3.

8.2.1 Eurocode 3 static verification results

The Eurocode 3 criterion II (equation 8.2) was found to have a maximum value of 158 MPa which yields a safety factor of 3 relative to the limit of 475 MPa (8.2). Since this criterion was always found to be a secondary design aspect, it is excluded from the results presented in this thesis.

The results for Eurocode 3 criterion I (equation 8.1) are presented in table 8.1. The proposed weld configuration has a partial penetration of 6.65 mm. To better understand the influence of penetration depth, a 8.25 mm and a full penetration (12 mm) were also included in the evaluation. Looking at the variation in the results along the length of the dump (from W1 upstream to W12 downstream), the critical welds are located towards the center of the TDEs. This is in good agreement with the longitudinal expansion of the TDEs defined in section ?? applying the highest axial stresses to the welds located in the center of the TDEs. Taking into account a penetration depth of 6.65 mm, the stress component product of weld 8 is 392 MPa resulting in a safety factor of 1.3 with respect to the limit of 528 MPa. The safety factor is low considering the fact that the effect of stress concentrations is excluded from the assessment. An increase in weld depth to 8.25 mm results in a safety factor of 1.7. The full penetration weld where it is reasonable to assume no effects of stress concentrations has a safety factor of 2.4 in the most critical weld.

While the results are compliant with Eurocode 3 on static verification, they rely on assumptions that can have a high influence on the results. Given the high stress levels, it is immediately apparent that the safety factor for fatigue would be low. To partially mitigate this problem, an 8.25 mm partial penetration weld is selected as the baseline for further assessment. A new weld qualification process needed to be triggered, given the new geometry and the almost 20-year period that elapsed since the last time these welds were made.

Weld cm from USF	W1	W2	W3	W4L	W5L	W6L	W7	W8	W9	W10	W11	W12
	14	49	123	210	297	385	472	545	617	690	763	836
6.65 mm												
Scaled 1.80	83	86	193	271	314	353	337	392	281	192	156	69
Safety factor	6.4	6.1	2.7	1.9	1.7	1.5	1.6	1.3	1.9	2.8	3.4	7.7
8.25 mm												
Scaled 1.45	67	70	156	219	253	285	272	317	227	155	126	60
Safety factor	7.9	7.6	3.4	2.4	2.1	1.9	1.9	1.7	2.3	3.4	4.2	9.4
Full pen.												
Scaled 1.00	46	48	108	151	175	197	188	219	157	107	87	40
Safety factor	11.4	11.0	4.9	3.5	3.0	2.7	2.8	2.4	3.4	4.9	6.1	13.7

Table 8.1: Eurocode 3 EN 1993-1-8 static design resistance of TDE welds. Welds are located longitudinally by their distance from the upstream window. Lowest safety factors is in the center area of the TDEs due to longitudinal expansion. Weld 8 is the most critical weld. **Criteria limit:** 528 MPa.

8.3 Welding Performance Qualification Record (WPQR) according to ISO15613 of Uranus 45

Additional studies need to be performed to better assess the stress concentration factor in the welds and their behaviour upon beam impact. Similarly to the case of the downstream flange, the response of the weld can be assessed more precisely by using a sub-modelling approach that allows for finer meshing of the weld geometry.

A weld qualification was performed by the EN-MME section on flat test specimens [13]. The weld geometry was adapted from the drawings in figure 8.1 and the weld was performed using representative procedure for the welds in the TDEs.

The specimens were metallographically examined to detect potential defects in the welds. No defects were found. Figure 8.6 shows the metallographic examination obtained via macro etch testing. A series of indentation tests on the cross section revealed no decrease in hardness between base material, heat affected area and weld seam.

Furthermore, a total of 4 samples were cut for tensile testing. Conforming to the relevant norms, these samples had a full penetration geometry with a thickness of 8.25 mm. The specimens consistently failed far from the welding area, pointing to a higher strength of the weld than the one of parent material. The results are summarised in table 8.2.

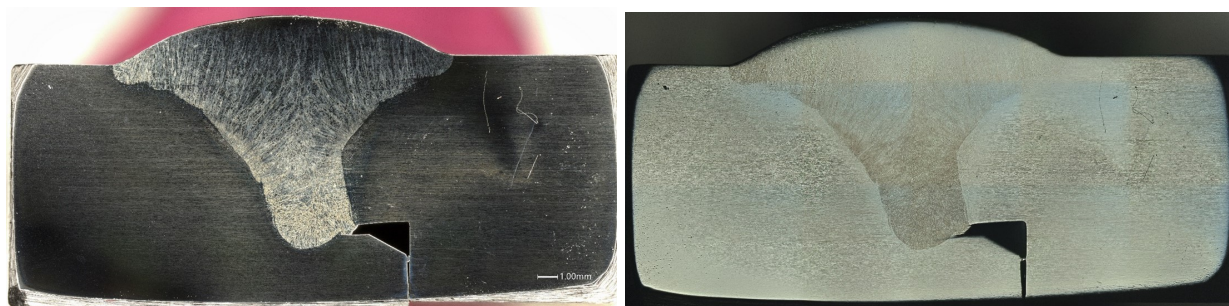


Figure 8.6: Destructive metallographic examination of partial penetration welded specimens of SS318LN

	Yield strength [MPa]	Ultimate tensile strength [MPa]
Material certificate	Min. 480	Min. 680
Specimen 1	616	724
Specimen 2	576	725
Specimen 3	590	724
Specimen 4	578	722

Table 8.2: Results of tensile testing of 8.25 mm-thick welded specimens of 318LN.

Besides being necessary for a complete qualification of these welds, these test provide valuable input for their simulation. Since the strength of the weld is proven to be stronger than the parent material, it is reasonable to model the weld geometry as a single body with uniform material properties in the weld seam and parent material. Furthermore, the visual inspection provides information on the actual weld geometry that can be directly implemented in its model.

8.4 Tensile Tests Simulation of Weld

With the weld qualification as the base, several 2D tensile tests were simulated to better assess the effect of stress concentration. The image from the Macro examination was digitized to accurately adopt the weld geometry.

As shown in 8.7, the models use 2D plane stress elements. The length in the longitudinal direction is of 5 cm to avoid close proximity between the area of interest and the boundary. The results are scaled to a thickness of 1 cm. A strong mesh refinement is imposed in the area of the weld, and especially around the fillets.

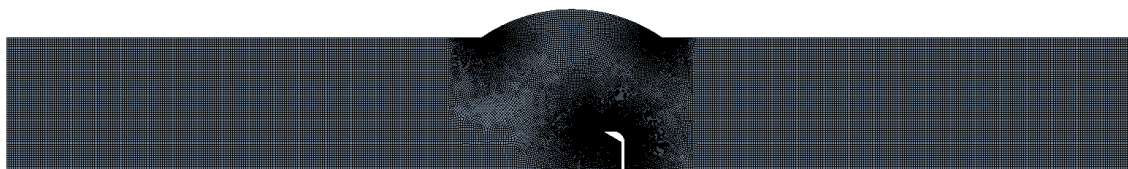


Figure 8.7: Mesh of the 2D model for tensile simulation. The model is fixed to the right boundary and a force of 5 kN is applied to the left boundary.

The geometry can be seen after the meshing in figure 8.8. Here, the mesh of the global model is overlayed in orange to give a sense of the element size difference between the two models. The simulations are set up in Ansys Mechanical® using a static structural module with an implicit algorithm.

For the tensile simulation, a force of 5 kN is applied to the boundary. The initial purpose of the simulations is to assess the possibility of reproducing a similar stress distribution to the one in the TDE welds. The stress concentration factor is determined by comparing the von Mises stress along two paths defined in the models.

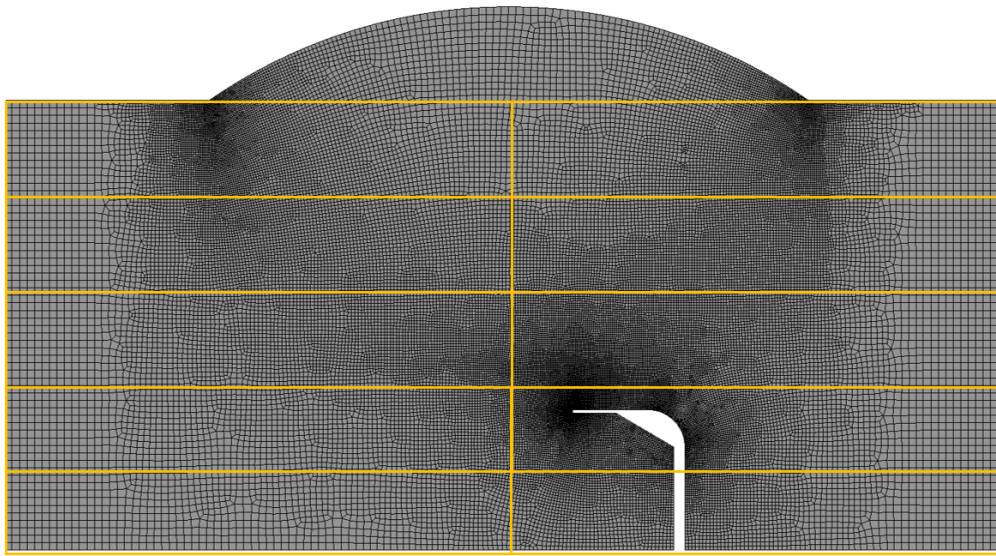


Figure 8.8: Mesh of the 2D model for tensile simulation. Overlaid orange squares represent element size in the global model.

- Weld path. In the longitudinal center of the area with reduced thickness.
- Reference path. Located in the parent material, far from the welded area and from the boundary.

Two orange circles on the same figure highlight areas that are subject to stress concentration.

- Left circle. The actual geometry of this area cannot be controlled as it is formed as a result of the welding. As seen in figure 8.6 where the actual geometry in this area is difficult to determine. For this reason, it is difficult to evaluate using FEM and is therefore left out at this stage.
- Right circle. The geometry of this area can be controlled as it is fully included in the parent material. Its optimization will be shown in the following.

Six different geometries were simulated to understand the impact of different modifications. These can be seen in figure 8.10.

- The original weld geometry.
- A configuration in which the weld seam is machined flush after welding.
- A configuration in which the right inner fillet is increased to reduce stress concentrations.
- A configuration with a partial penetration of 6.65 mm instead of 8.25 mm.
- Two reference simulations with a uniform thickness of 12 mm and 8.25 mm were performed for comparison.

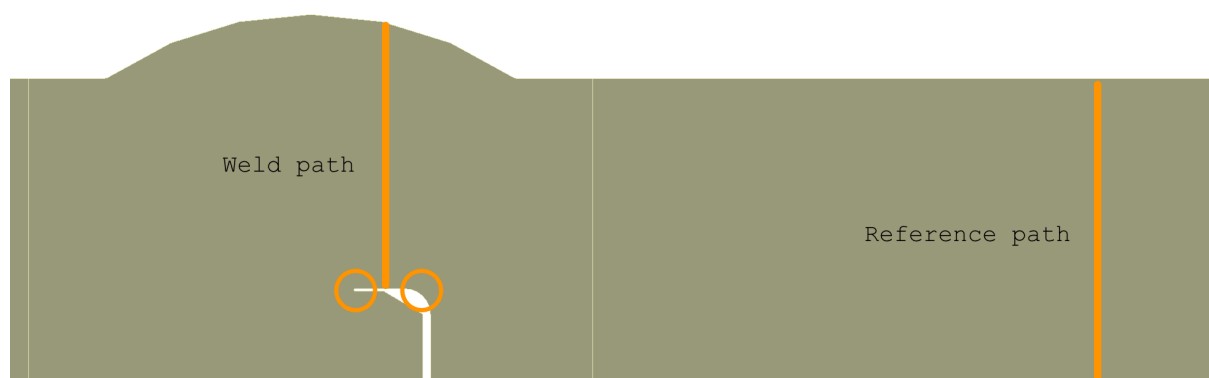


Figure 8.9: Paths in the model used for evaluating the stress concentration factor of the partial penetration weld.

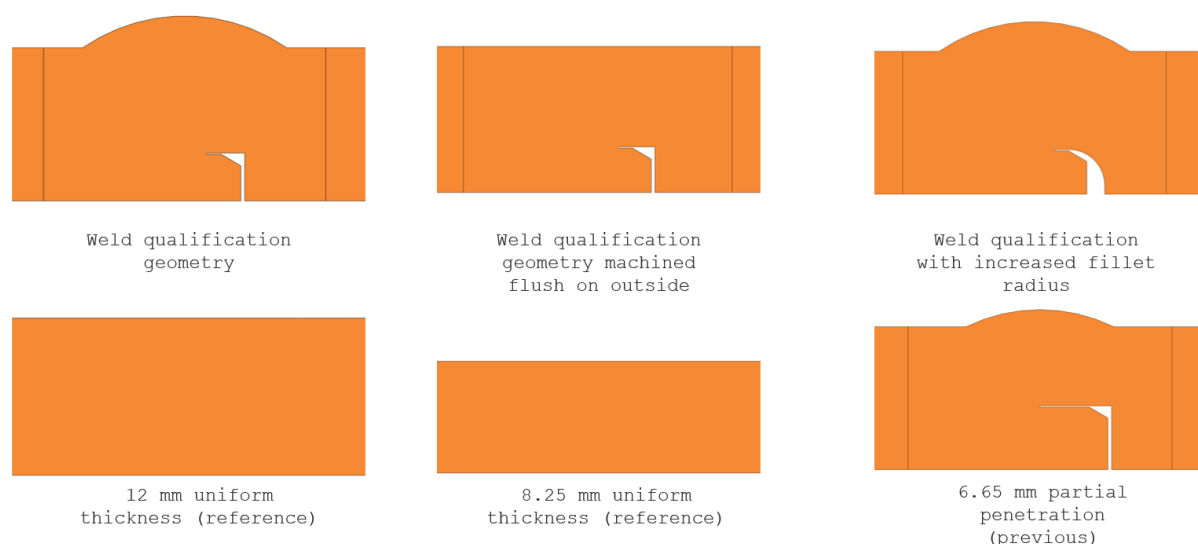


Figure 8.10: Paths in the model used for evaluating the stress concentration factor of the partial penetration weld.

8.4.1 Tensile Tests Simulation Results

Looking at figure 8.11 the von Mises stress on the reference path is uniform through the thickness at 41.7 MPa. This is consistent between all models with 12 mm thicknesses on the reference path. As such, it can be concluded that the reference path is not affected by the weld geometry. The weld path in figure 8.11 shows a non-uniform von Mises stress with a strong increase towards the weld root. This is in good correspondence with the stress flow line analogy. Due to the abrupt thickness transition, the stress flow lines that were distributed uniformly over the thickness at the reference path are in the weld path squeezed together, resulting in an increase in stress levels. In figure 8.11 the ratio between the stress levels in the reference path and the max stress in the weld path is 3.7 - from 41.7 MPa to 155 MPa. This means that simply scaling the results with the corresponding area reduction factor, in the static Eurocode 3 verification in section 8.2.1, is an assumption that is misleading by a factor 2. Scaling stress levels with 1.45 for an 8.25 mm partial penetration weld does not capture the effect of stress concentration. The tensile simulation showed that a more correct interpretation of the stress increase factor is 3.7, which accounts for both the area reduction and the stress concentration factors.

CHAPTER 8. CIRCUMFERENTIAL WELDS

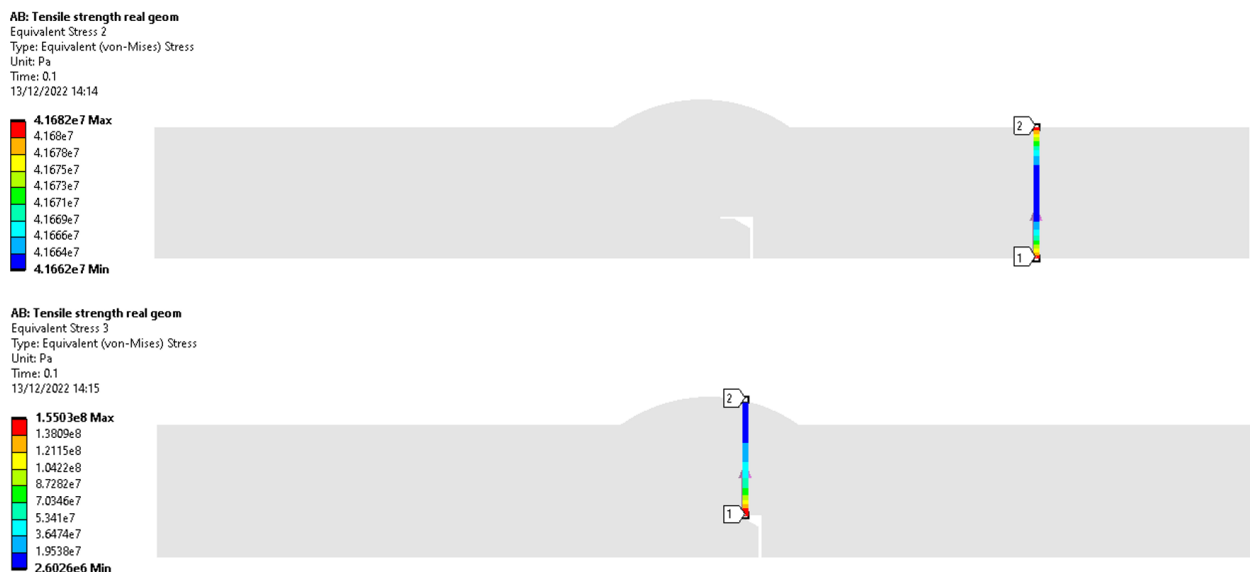


Figure 8.11: Results of a tensile test of a weld geometry. Note that reference von Mises stress are uniform through thickness. Von Mises stress in weld path is non-uniform with concentration towards the bottom of the weld path.

Several other interesting observations can be derived from table 8.3. Applying the same 5 kN load to a 12 mm and an 8.25 mm geometry with uniform thickness results in a stress increase of 1.45 from 41.7 MPa to 60.6 MPa consistent with the assumption in section 8.2.1. When decreasing the weld penetration depth to 6.65 mm the increase in stress is a factor of 5.5, which means that decreasing the depth of the weld scales the stress levels non-linearly.

Tensile simulations	Reference	Weld max.	Weld min.	Factor	Factor
Load: 5 kN	[MPa]	[MPa]	[MPa]	max.	min.
Weld quali. original geometry	41.7	155	2.6	3.7	0.1
Weld quali. flush geometry	41.7	160	0.6	3.8	0.0
Weld 6.65 mm penetration	41.7	230	10.9	5.5	0.3
12 mm uniform thickness	41.7	-	-	-	-
8.25 mm uniform thickness	60.6	-	-	-	-

Table 8.3: 2D Tensile simulation results of different weld geometries.

As previously stated, the fillet on the right side of 8.12a, can be optimised as it lies fully in the parent material. A sensitivity study on various fillet radii is presented in figure 8.12. As can be seen, an increase in fillet radius exhibits a diminishing returns type of behaviour. Where an increase in fillet radius from 0.04 mm to 0.5 mm reduces the stress levels from 450 MPa to 280 MPa, an increase from 2 mm to 3 mm has a much smaller impact.

While the tensile simulations provide a good understanding of the stress concentration factors due to the weld geometry, they do not provide any information on the stress levels in the weld when loaded with the response of the TDEs upon beam impact. Therefore, a number of simulations were performed using a submodeling approach with the aim of estimating the stress level due to the weld. These will be presented in the following sections.

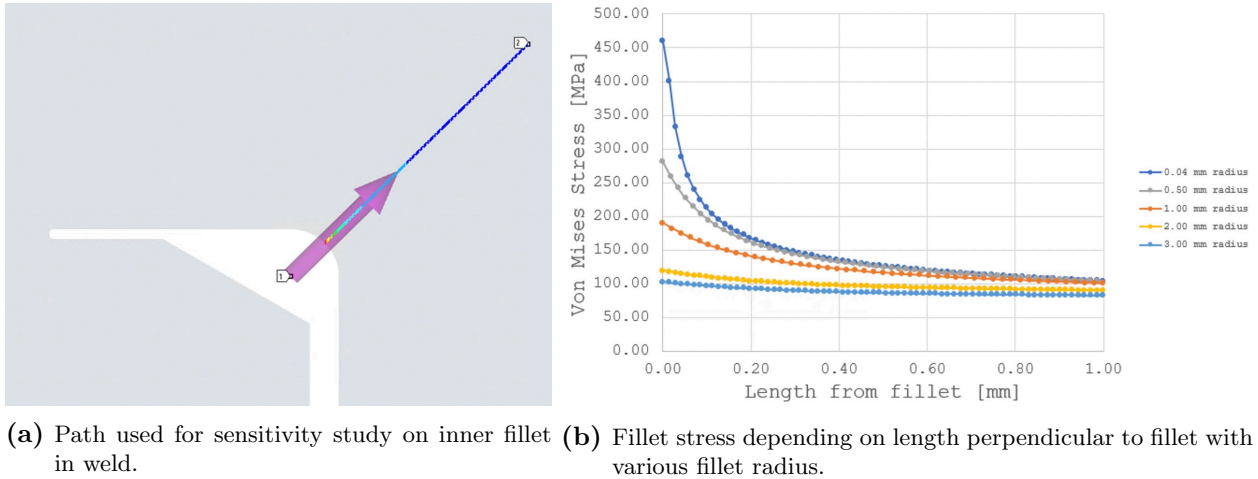
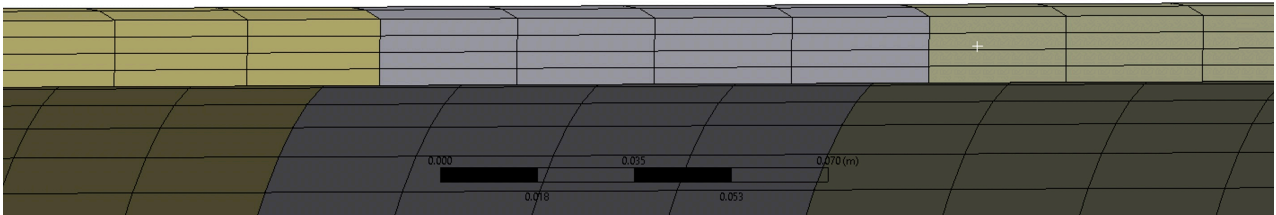


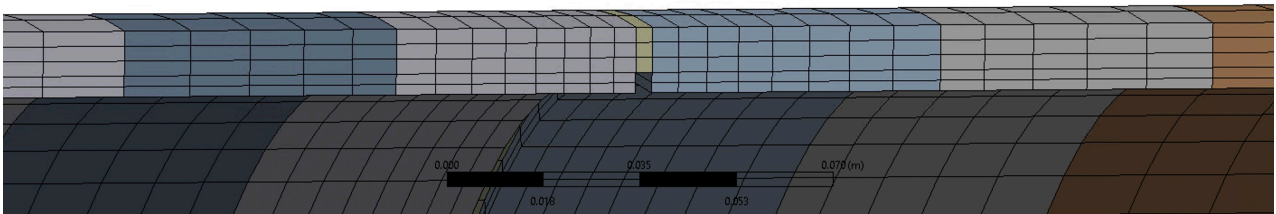
Figure 8.12: Sensitivity study of inner weld fillet radius.

8.5 3D Submodel Simulation of Weld

For a submodel to yield accurate results, the stiffness of the global model and submodel must be within a fine tolerance. Since the global model of the TDE is modelled with a uniform thickness at the weld locations, this criterion is not met. For longitudinal stiffness, a 4.3 % difference was found when a realistic geometry was considered. More importantly, the weld geometry is expected to largely affect the rotational stiffness due to the reduced thickness in the weld. Therefore, the global TDE model was re-run with a modified geometry of the vessel. Cutouts representing the reduced thickness in the welds were implemented. Additionally, the mesh was refined from 4 to 5 divisions through the thickness of the vessel, in order to better capture gradients, and a proper growth rate in element size toward the welds was implemented. A cross-section of the vessel showing the cutout due to the weld can be seen in figure 8.13b and a comparison of the original mesh and the refined mesh can be seen in figure 8.13.



(a) Cross-section of the original mesh of the TDE vessel at a weld location.



(b) Cross-section of the updated mesh of the TDE vessel at a weld location to better capture longitudinal - and rotational stiffness.

Figure 8.13: Mesh modifications of the global model of the TDE to prepare for sub-modelling.

With the boundary conditions more accurately capturing the behaviour of the TDE in the weld locations, a 3D submodel was established. The same geometry as the tensile simulations obtained from the weld qualification is used. Instead of fixing the model on one end and applying a uniform load to the other end, as in the tensile simulation, the submodel is constrained by the displacements at the boundaries obtained from the TDE simulation. The boundaries are located 5 cm on each side of the weld.

The initial approach was a 3D submodel in LS-Dyna[®] based on the explicit algorithm. Using this method, the displacements on each boundary of the weld would be captured and transferred to the submodel, meaning that not only the longitudinal expansion of the TDE would be applied, but also the radial expansion and flexural deformation. While this would accurately capture the boundary conditions of the submodel, the mesh fineness due to the explicit algorithm stability condition was largely limited.

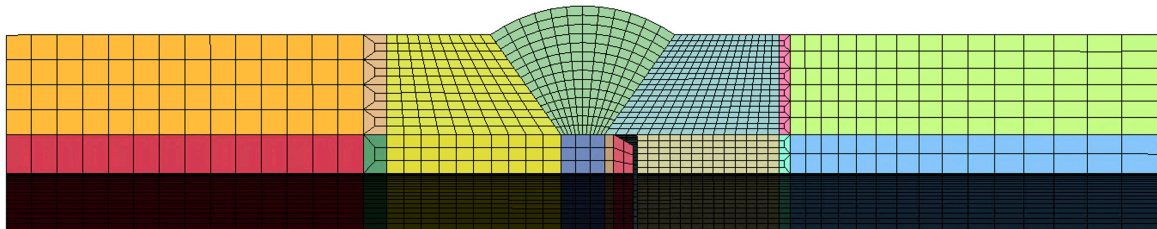


Figure 8.14: Mesh of weld used for 3D submodel in LS-Dyna[®]

In figure 8.14 the element size of the mesh is seen, highlighting the coarseness of the mesh when compared to figure 8.8. The model contains 530 000 elements and 460 000 nodes and is solved in 9 hours on 64 cores using MPP. Despite the moderate number of elements and nodes, the characteristic length of the elements substantially increases the number of timesteps to meet the CFL stability condition of the explicit algorithm. The coarse mesh did not accurately capture the stress gradients, and therefore it was concluded that the explicit algorithm was not suited for the application.

8.6 2D Axisymmetric Submodel Simulation of Weld

The next approach was to use a 2D axisymmetric model using an implicit algorithm. Using this method, the same mesh as used for the tensile simulations seen in figure 8.8 is adopted, allowing for detailed meshing of the weld geometry properly capturing the stress development.

Changing the simulation from a 3D to a 2D axisymmetric simulation has two key impacts:

- The boundary displacements are not transferred to the model in a 3D domain. Indeed, the displacements at the boundaries of the 2D axisymmetric model are interpreted as being uniform around the circumference. Moreover, displacements in the 2D axisymmetric simulation can only be applied axially and radially. The radial displacement of the global model, however, is the sum of its radial expansion flexural deformation, which cannot be captured by an axisymmetric simulation. Since these responses are secondary for the stress levels of the welds, they were left out, meaning that the submodel only is applied axial displacements from the global model. This concept is better illustrated in figure 8.15.

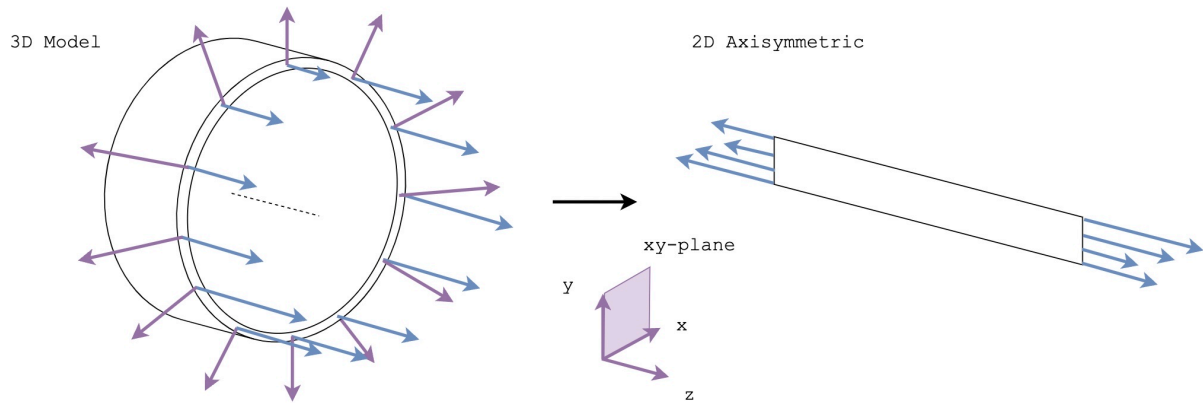


Figure 8.15: Visualization of boundary displacement simplification. The 3D model captures displacement variations in all directions along the circumference of the vessel. The 2D axisymmetric model is applied axial displacements varying through the thickness as a function of the time domain but uniform around the circumference.

- The energy is applied in the YZ-plane and is considered uniform around the circumference despite the unsymmetrical sweep of the beam.
- The resolution of the displacements is reduced since Ansys[®] Mechanical cannot read the displacement file from LS-Dyna[®] and therefore the resolution is 10 times less frequent. This issue was mitigated by increasing the output frequency of the global model, especially up to the first complete longitudinal oscillation.

The axial displacements at the boundaries are extracted from LS-Dyna[®] and applied to the model in Ansys[®] using an APDL script. The displacements are captured for all six nodes through the thickness of the vessel at both boundaries in the global model and tabulated. By building a table with displacements through the thickness of the vessel as a function of time, Ansys[®] can linearly interpolate in both position and time. This approach is described in more depth in A.8.

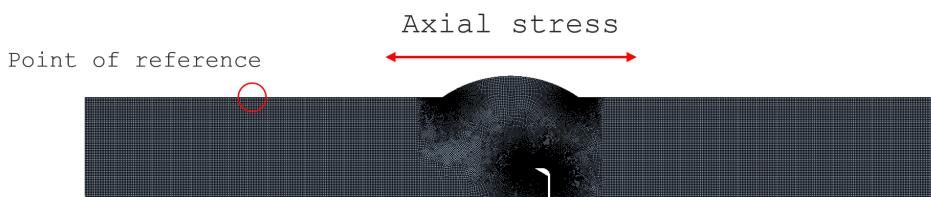


Figure 8.16: Illustration of point of reference for comparison of axial stress between global model and sub-model

To validate the model, a reference element far away from the weld geometry was compared between the global model and the sub-model. The point of reference is shown in figure 8.16. Comparing the axial stress evolution in the same reference element of the global model and sub-model indicates whether the weld geometry is subject to consistent loading conditions. In figure 8.17 the comparison of the axial stress is seen, showing a good correspondence between the global model and the sub-model. This means that the boundary displacements and energy deposition are consistent between

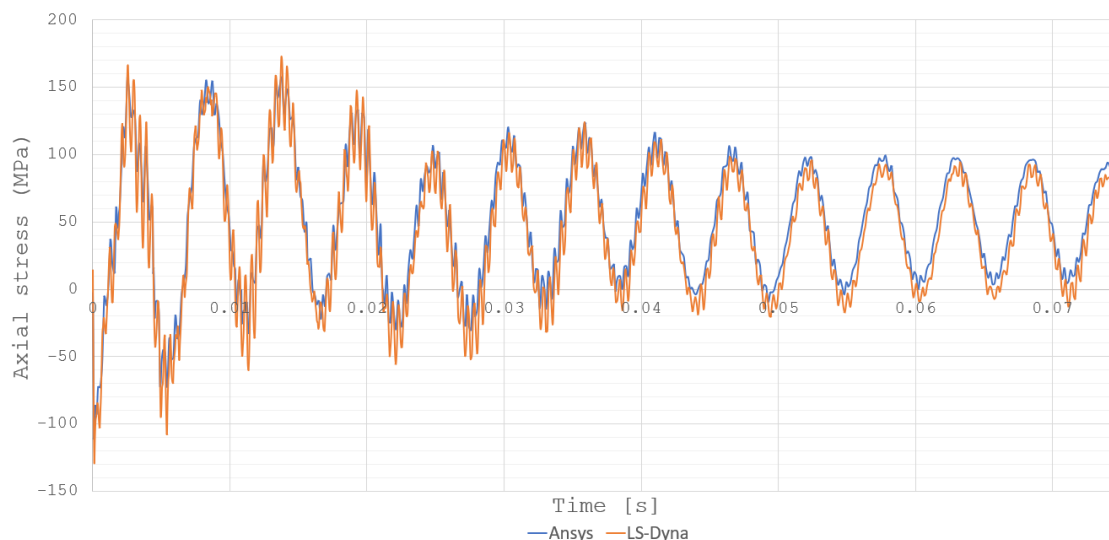


Figure 8.17: Comparison in axial stress between global model and 2D axisymmetric sub-model.

the 3D simulation of the global model, and the 2D axisymmetric simulation of the weld. Therefore, the results obtained with this new submodeling approach are deemed representative.

8.6.1 2D Axisymmetric Weld Submodel Results

Figure 8.18 shows the 2D axisymmetric weld geometry at the time step where the maximum equivalent von Mises stress is reached. It can be observed that the weld is stretched due to the displacements applied to its boundaries. Due to its geometry, this results in the weld bending inward toward the core. As a result, the fillets are exposed to high stress levels. A maximum von Mises stress at 482 MPa is reached on the fillet on the left side of the weld. Although this level of stress is less than the yield strength of the weld obtained in the weld qualification in table 8.2, as previously stated, the results in this area are strongly dependent on the geometry that is produced during welding, that is difficult to control and poorly reproducible. For this particular case, the result is obtained considering a small fillet having a 0.04 mm radius.

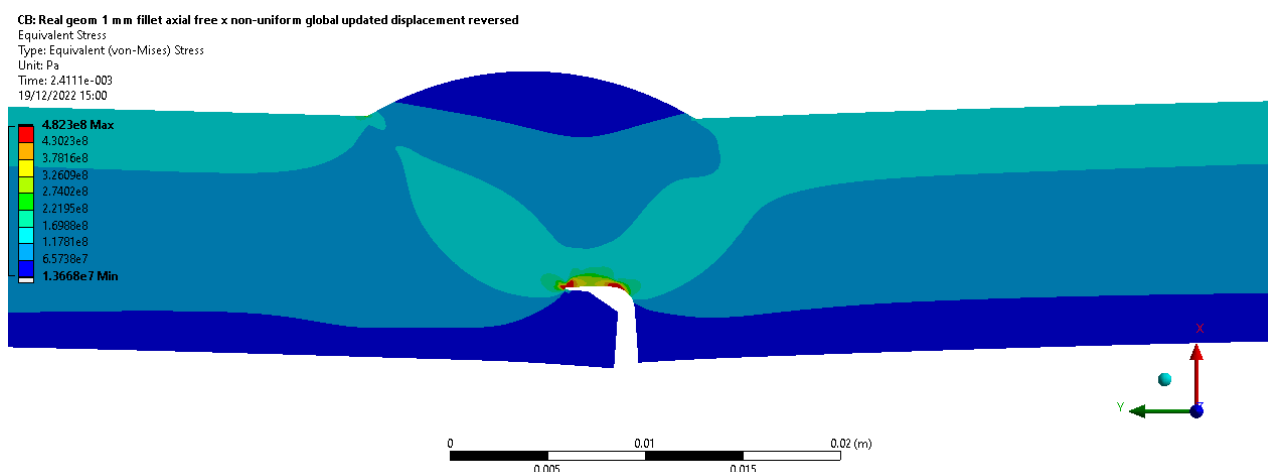


Figure 8.18: Maximum equivalent von Mises stress in weld geometry. Deformation scaled by a factor of 60 to highlight the deformation profile.

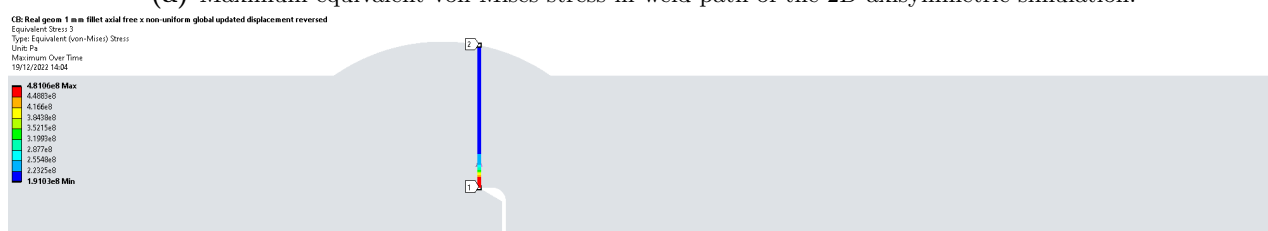
A stress concentration factor of the weld was evaluated similarly to the previous approach used for

CHAPTER 8. CIRCUMFERENTIAL WELDS

the tensile tests in 8.4.1. In figure 8.19 a comparison in equivalent von Mises stress in the weld - and reference path is seen. The results are obtained at the timestep where the maximum von Mises stress in the weld path was reached. The maximum von Mises stress in the reference path is 189 MPa and the minimum 156 MPa yielding an average of 173 MPa. Compared to the weld path, the maximum von Mises stress is 481 MPa. This corresponds to a stress concentration factor between 2.5 and 3.1.



(a) Maximum equivalent von Mises stress in weld path of the 2D axisymmetric simulation.



(b) Equivalent von Mises stress in reference path of the 2D axisymmetric simulation at same time step as figure 8.19a.

Figure 8.19: Stress concentration factor in 2D axisymmetric weld submodel simulation.

Recalling the results from table 8.4.1 the weld geometry was found to yield a stress concentration factor of 3.7, which means that the 2D axisymmetric submodel is less exposed to the effect of stress concentration in the weld geometry. This is most likely due to the fact that the submodel captures variations in axial displacements through the thickness at the boundaries, whereas the tensile models were applied uniform force at the boundaries.

8.7 Conclusion

The proposed weld configuration for the manufacturing of the Run 3 spares included 12 circumferential welds along the length of the dump with a partial penetration depth of 6.65 mm. The partial penetration configuration is implemented, despite introducing stress concentration factors, to protect the shrink-fitted graphite blocks and enable the implementation of a male-female configuration to precisely align the segments upon welding.

A Eurocode 3 static verification was performed using the simulation results obtained from the Target Dump Externals upon beam impact to highlight the most critical welds. The assessment resulted in a static validation of all welds in the model and an increase in partial penetration depth from 6.65 mm to 8.25 mm was adopted. However, the Eurocode static verification was based on assumed stress scale factors of simulation data to represent a partial penetration weld.

A weld qualification showed strength performance of the partial penetration weld higher than the parent material. This allowed the sub-modelling of the weld geometry using a single body with uniform material properties. Furthermore, the weld qualification provided a visual inspection of the weld seam adopted in submodelling approaches for precise representation.

CHAPTER 8. CIRCUMFERENTIAL WELDS

The tensile simulations estimated a stress concentration factor of up to 3.7. This is a factor 2 higher than the area reduction factor of 1.8 that was considered in the Eurocode 3 verification for a 8.25 mm partial penetration weld. Tensile simulations of different weld geometries provided an understanding of the stress level impact of different modifications, resulting in a fillet radius increase to 1 mm on the inner face of the parent material.

While the tensile simulations provided an understanding of stress concentration factors in the weld geometry, they did not describe the level of stress in the weld due to the vibration of the TDEs. To assess this, a 3D sub-model, constrained by displacements from the global model was adopted to closer represent the loading case of the welds. However, the stability conditions of the explicit algorithm prevented the implementation of a high-resolution mesh due to computational cost, and therefore the 3D sub-model was replaced by a 2D axisymmetric sub-model using the implicit algorithm. The 2D axisymmetric model illustrated the deformation profile and yielded stress levels close to the yield strength of the material at 482 MPa and a stress concentration factor of 3.1.

The fatigue resistance of the weld was not captured inside the timeframe of the work related to this thesis. The fatigue aspect is expected to be important, due to the weld geometry and the vibrations of the Target Dump Externals upon beam impact, and therefore several approaches for assessing this matter are discussed in the following chapter.

Discussion

The finite element analysis is a powerful numerical method that yields approximate solutions for components with complex geometry and loading conditions. In this thesis, finite element algorithms were used for different applications, highlighting the advantages and limitations of explicit and implicit codes. The explicit algorithm excelled in simulations of dynamic responses upon impact of the beam, where a small time step ensured that the behaviour of the components was captured accurately. The implicit code was used for applications where a fine mesh was required to capture the impact of geometry details, but a small time step was of less importance.

Despite the simulations of the TDEs upon beam impact showed good correspondence with experimental data, the precision of the models is always depending on the engineer's capability of describing reality. Several simplifications in the simulation of the TDEs upon beam impact were made. For example, it was chosen not to include the support frame and mounting system in the model, as well as only applying energy deposition to the vessel, flanges, and windows, and not the core. Furthermore, the approximate solution will always be limited to the resolution of the mesh where the density of nodes passing and receiving the information is of great importance. It is absolutely critical to obtain a sense of the impact of the simplifications and assumptions made when using finite element analysis, enabling one to balance the quality of the simulation result with the number of hours spent to implement loading conditions that represent reality as closely as possible.

While submodeling has proven to be a powerful tool to assess the response of small details in such a large model, it is understood that the quality of these results are limited to the quality of the global model. Furthermore, this process entails the strong use of interpolation over time and space for the application of the boundary conditions to the submodels and, as such, should be checked carefully.

While the assessment and improvements of the downstream flange were clear, the assessment of the weld configuration still needs further work before final conclusions can be drawn. The work contributed to a much better understanding of the weld configuration through a novel interdisciplinary assessment. Additional evaluation must be performed to assess the response of the welds upon beam impact and over their lifetime. In particular, a tensile test specimen was shown to be strongly representative for the load case to which the welds are subjected in operation. As such, a tensile test of the realistic partial penetration weld is suggested to benchmark the results of the simulations and assess

CHAPTER 9. DISCUSSION

their reproducibility. Once this is done, a similar combined assessment should be used to evaluate the fatigue life of the welds.

Conclusion

This thesis aimed to validate the upgrades of the downstream flange and the weld configuration for the manufacturing of the TDE Run 3 operational spares. With the use of explicit finite element code in LS-Dyna®, the upgrades were evaluated and updated to comply with the design criteria defined at the beginning of the thesis.

The results of a multistep weakly coupled thermomechanical simulation of the Target Dump Externals upon beam impact showed good correspondence with experimental data.

The use of a submodeling approach allowed detailed meshing and loading of the downstream flange to better understand the behaviour of the component upon beam impact. This highlighted critical areas of the design. A study of several downstream flange candidate designs revealed that the proposed flange design for the Run 3 spares had similar performance to that of the currently operational dumps, with stress levels close to the material's yield strength. Based on the results obtained during the evaluation, an updated design was suggested, resulting in a reduction of 42 % in the stress peak. With the use of Laser Doppler Vibrometer data obtained from the currently installed dump, a damping ratio of the system was found, allowing a realistic multiaxial fatigue assessment to be performed. The updated design showed infinite life, ensuring compliance with the requirement of 1600 full-intensity beam dumps of 539 MJ during its lifetime.

An interdisciplinary assessment of the partial penetration weld configuration provided a valuable understanding of the stress development within the weld geometry of the Target Dump Externals. A Eurocode 3 static verification was performed using simulation data to highlight the most critical weld locations. However, the verification was based on simple stress scaling by the ratio of the areas. This assumption was found to be not conservative by implementing the geometry of the weld qualification in the simulation of a tensile test. The weld qualification showed a higher strength of the weld than the parent material, allowing sub-modeling of the weld geometry to be performed as a continuous body without contacts and with uniform material properties. A 2D axisymmetric sub-model of the weld geometry revealed stress concentration factors up to 3.3 with respect to the stress level in the parent material yielding von Mises stress close to the yield strength of the material. As a response to the findings, the partial penetration depth of the weld was increased from 6.65 mm to 8.25 mm.

CHAPTER 10. CONCLUSION

Finally, a discussion of the methods used to assess the downstream flange and weld configuration in this thesis was presented, highlighting limitations, potential improvements and suggestions for further work.

Appendix

A.1 Equivalent Beam Energy Cases

The beam energy is 539 MJ during Run 3. The following document in Mathcad calculate equivalent energy cases to grasp the energy amount. Material properties of copper found on The Engineering Toolbox [3] [2] and information on airplane weight found on Pilot Teacher [1].

Beam energy equivalent calculations

$$E_{beam} := 539000000 \text{ J}$$

Copper heating and melting:

$$c_{p_{copper}} := 0.385 \frac{\text{J}}{\text{gm} \cdot \text{C}} \quad L_{copper} := 206 \frac{\text{J}}{\text{gm}}$$

$$t_1 := 20 \text{ C} \quad t_2 := 1085 \text{ C}$$

$$E_{heatandmelt} = (c_{p_{copper}} \cdot (t_2 - t_1) \cdot mass) + (L_{copper} \cdot mass)$$

$$mass := \frac{E_{beam}}{c_{p_{copper}} \cdot (t_2 - t_1) + L_{copper}} = 874.964 \text{ kg}$$

539 MJ is the equivalent energy required to heat up and melt 875 kg

Kinetic energy:

$$mass_{vehicle} := 68000 \text{ kg}$$

$$E_{kinetic} = \frac{1}{2} \cdot mass_{vehicle} \cdot v^2$$

$$v := \sqrt{\frac{E_{beam}}{\frac{1}{2} \cdot mass_{vehicle}}} = 453.27 \frac{\text{km}}{\text{hr}}$$

539 MJ is the equivalent amount of kinetic energy stored in a 68 ton vehicle traveling at 453 km/h

A.2 Geometry Preparation and Simplifications of the TDE for Simulation

The geometry of the TDE is simplified to ease meshing. Geometry can be simplified as long as the overall stiffness of the model is maintained. Examples of geometry simplification with meshing are presented below.

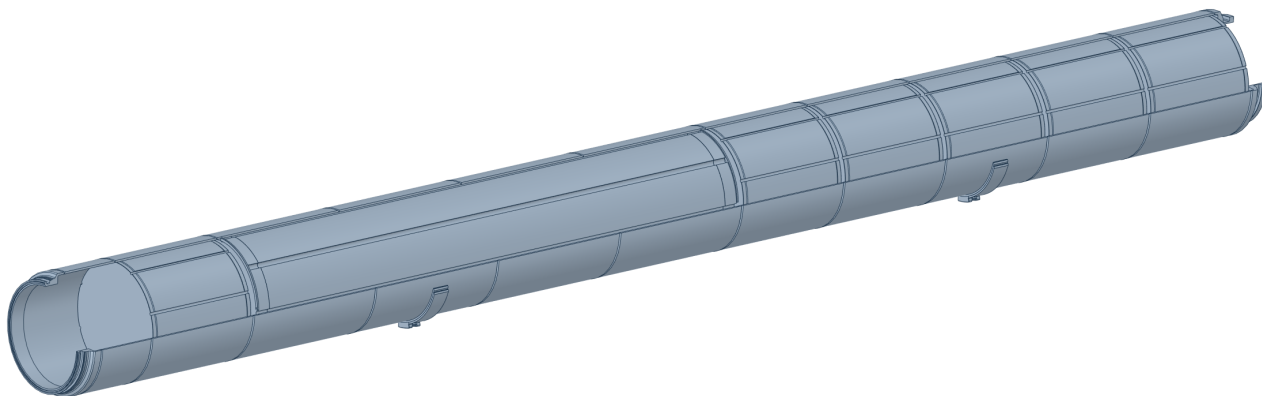


Figure A.1: Original geometry before simplification

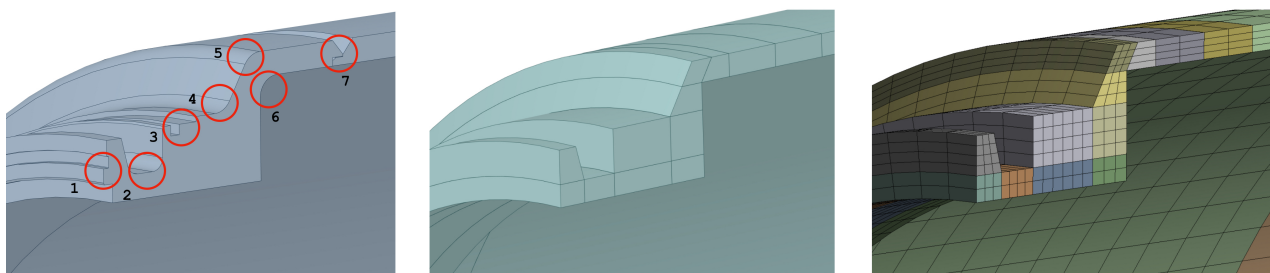


Figure A.2: Cross section of upstream flange. Left: original geometry. Center: simplified geometry with slices. Right: meshed. Simplifications include defeaturing (fillets, grooves and notches) and reducing weld geometry to a uniform thickness.

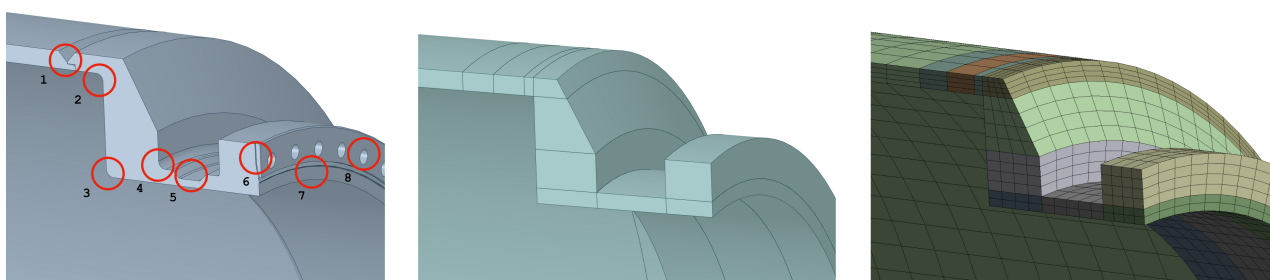


Figure A.3: Cross section of downstream flange. Left: original geometry. Center: simplified geometry with slices. Right: meshed. Simplifications include defeaturing (fillets, grooves and notches) and reducing weld geometry to a uniform thickness. Holes for window bolting are removed.

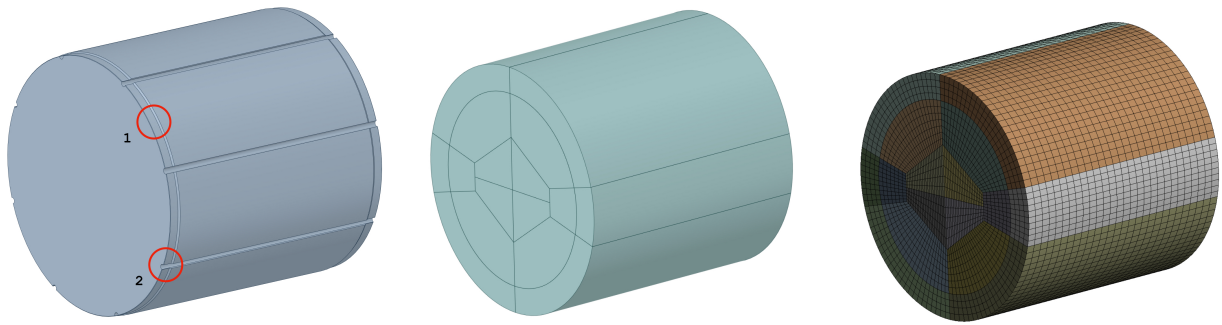


Figure A.4: High-density blocks. Left: original geometry. Center: simplified geometry with slices. Right: meshed. Simplifications include removal of grooves. Toward the ends of the high-density block a reduced diameter is removed making the total length of the block with uniform diameter.

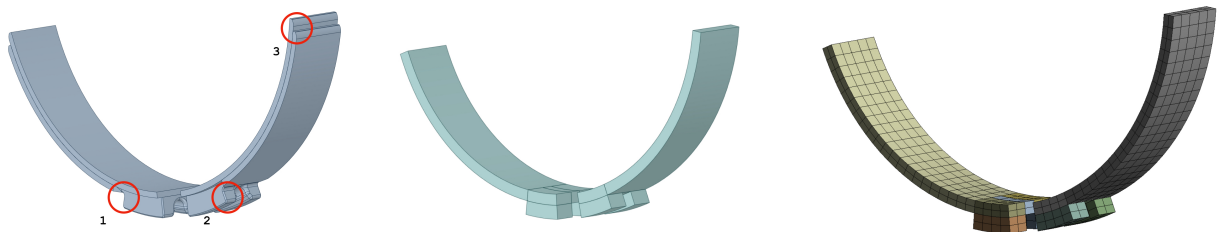


Figure A.5: Cableguide. Left: original geometry. Center: simplified geometry with slices. Right: meshed. Simplifications include removal of fillets and reducing complex geometry to bigger blocks of material.

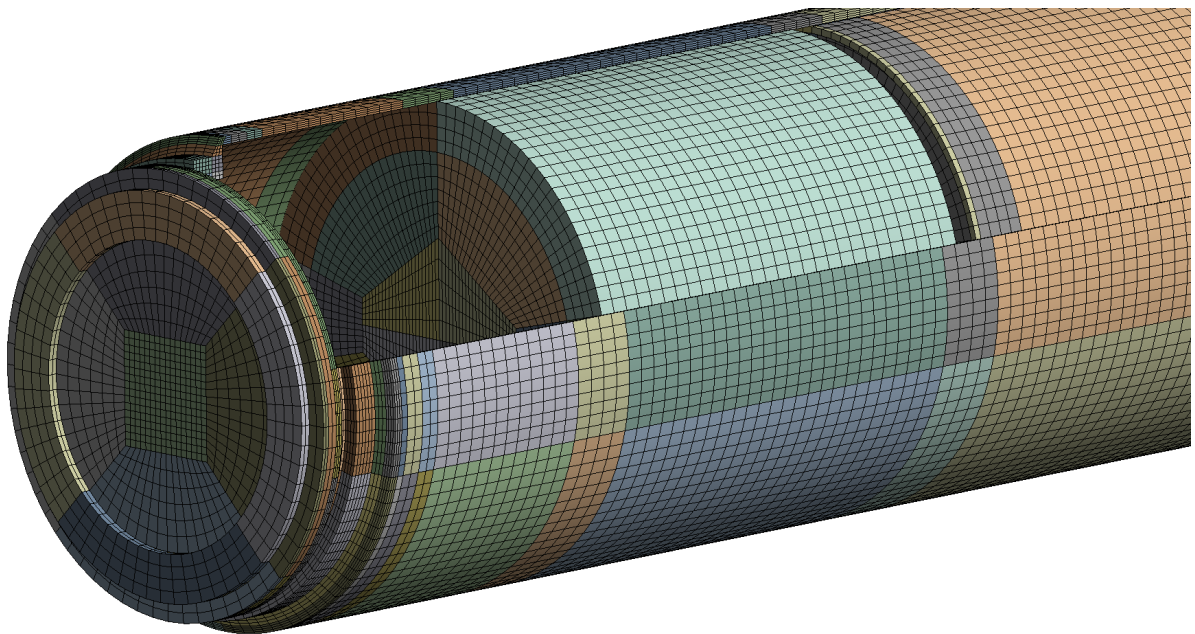


Figure A.6: Upstream part of TDE meshed.

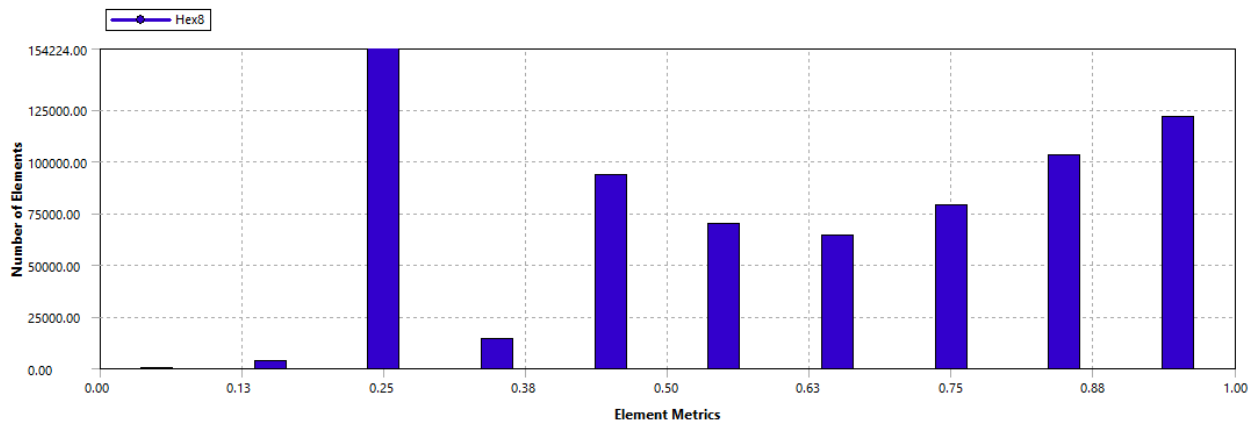


Figure A.7: Mesh metrics showing element quality where 1 is best and 0 is worst. An element with a quality of 1 is a perfect cube with equilateral edges. It is preferred to increase the number of elements through the thickness of the vessel. Due to the 12 mm thickness the elements in the vessel have low quality.

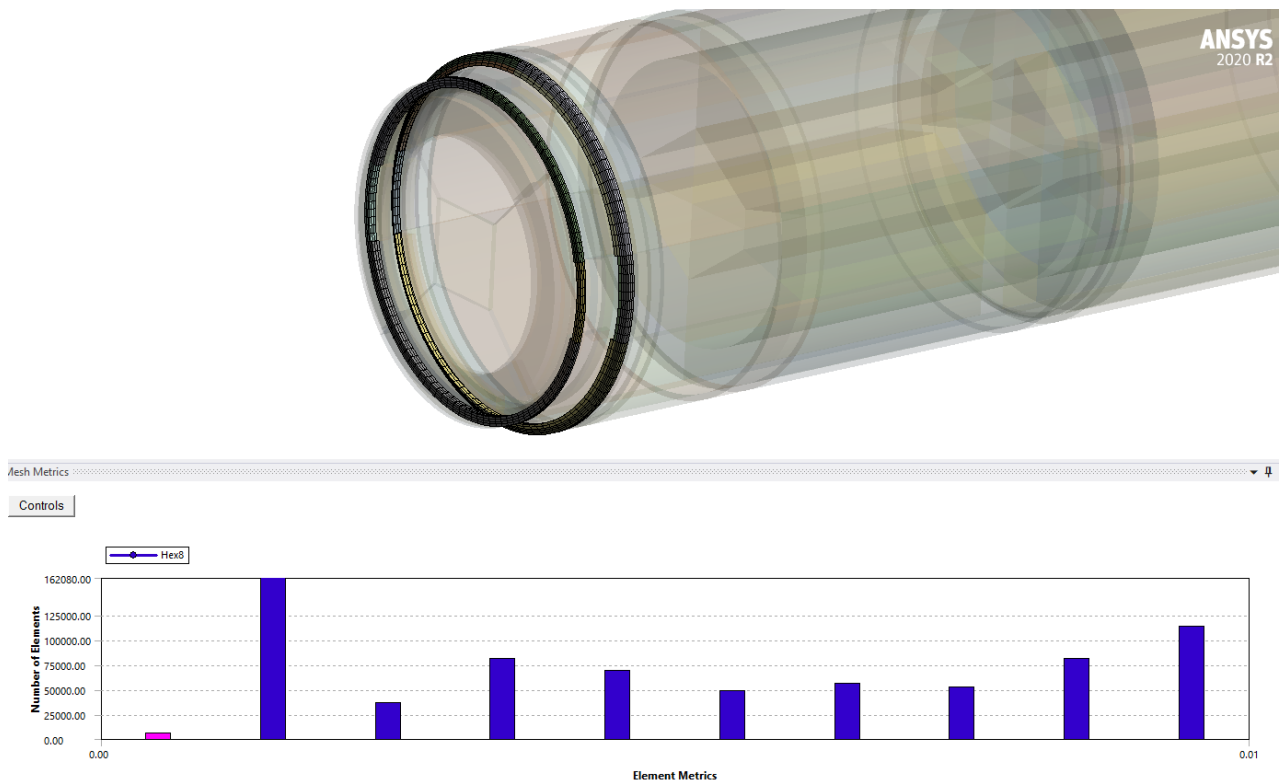


Figure A.8: Mesh metrics showing characteristic length of elements. Elements with the smallest characteristic length are highlighted and will be the elements defining the timestep size based on the CFL condition.

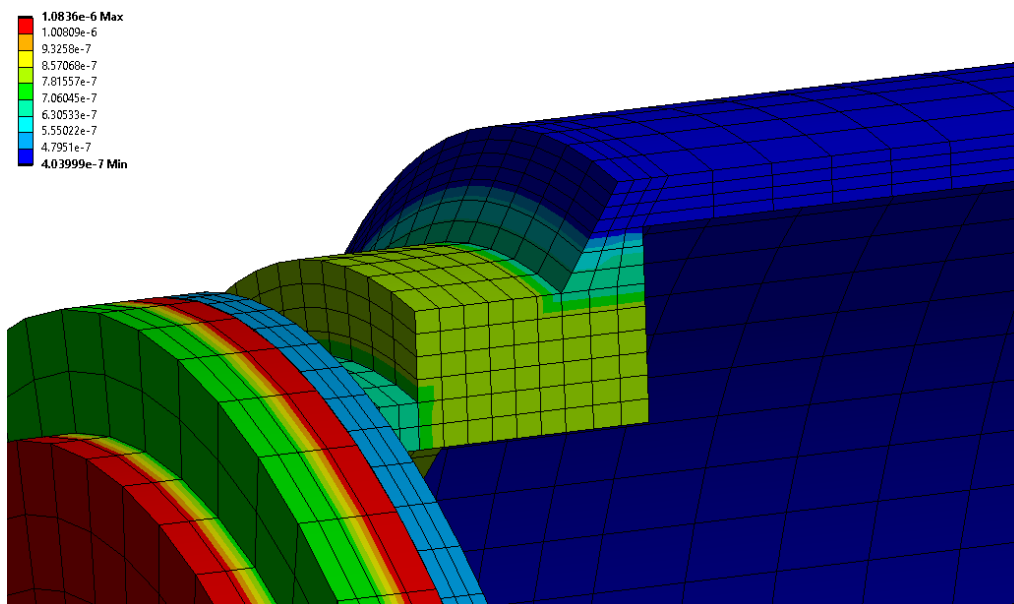


Figure A.9: Cross section of upstream part of TDE showing smallest element defining timestep size. Timestep size adopted is $4.04\text{e-}7$ second with a 0.9 safety factor. If the simulation is run for 1 second the solver will have almost 2 500 000 timesteps.

A.3 Meshing Guidelines

The FEA is a numerical method that calculate approximate solutions using a numerical method. The mesh is the solver input and therefore the result is heavily depending on the resolution and quality of the mesh. The following guidelines have been used in the meshing process to achieve a high performing mesh that yield correct results. The meshing guideslines are obtained from [31].

- Break up structures into simpler bodies to assist the meshing algorithm. Using shared topology the slicing of the bodies do not introduce contacts. With shared topology the sliced bodies of the same component share boundary nodes.
- Aim for at least three mesh layers through any given length, to capture bending stresses through the thickness of the components.
- Use small growth rate of neighbouring elements to ensure reduction of numerical errors due to abrupt transitioning.
- Defeature the geometry before meshing to avoid generation of small elements in fine detail geometry (fillets, grooves, notches, holes...).
- Use localized mesh controls to assist the meshing algorithm in transitional areas.
- Use hexahedral elements for simple geometries. Hexahedral elements are more precise than tetrahedral when considering simple geometries and are more economical.

A.4 Shrink Fitting Calculations (ANSI/ASME B4.1)

The simulation of the shrink fitting of the high-density core elements in the vessel is performed in LS-Dyna. To validate the result an implicit analysis in Ansys Mechanical was performed as well as hand calculations according to ANSI/ASME B4.1 [26]. The following shows the hand calculations.

Shrink fitting calculations

SIGRAFLEX 1.2 Core elements

Poisson's ratio of inner material	$v_i := 0.1$
Inner material inner diameter	$D_{i,i} := 0 \text{ m}$
Inner material outer diameter	$D_{i,o} := 0.7 \text{ m}$
Inner material elastic modulus	$E_i := 1.0 \cdot 10^{10} \text{ Pa}$

SS318LN Vessel

Poisson's ratio of outer material	$v_o := 0.31$
Outer material inner diameter	$D_{o,i} := 698.5 \text{ mm}$
Outer material outer diameter	$D_{o,o} := 0.7225 \text{ m}$
Outer material elastic modulus	$E_o := 1.9522 \cdot 10^{11} \text{ Pa}$

Calculations

Maximum interference $\Delta_i := D_{i,o} - D_{o,i} = 1.5 \text{ mm}$

Interference pressure
$$P_i := \frac{\Delta_i}{\frac{D_{o,i}}{E_o} \left(\frac{D_{o,o}^2 + D_{o,i}^2}{D_{o,o}^2 - D_{o,i}^2} + v_o \right) + \frac{D_{i,o}}{E_i} \left(\frac{D_{i,o}^2 + D_{i,i}^2}{D_{i,o}^2 - D_{i,i}^2} - v_i \right)} = 8.82 \text{ MPa}$$

Circumferential stress on outer due to interference
$$\sigma_{\theta,P} := P_i \cdot \left(\frac{D_{o,o}^2 + D_{o,i}^2}{D_{o,o}^2 - D_{o,i}^2} \right) = 261.19 \text{ MPa}$$

Von Mises Stress at outer Surface

$\sigma_r := -P_i \quad \sigma_z := 0 \text{ Pa} \quad \tau := 0 \text{ Pa} \quad \sigma_{\theta,cfq} := 0 \text{ Pa}$

$$\sigma_{VM} := \sqrt{\frac{(\sigma_r - \sigma_{\theta,P} - \sigma_{\theta,cfq})^2 + (\sigma_{\theta,P} + \sigma_{\theta,cfq} + \sigma_z)^2 + (\sigma_r - \sigma_z)^2}{2}} + 3 \tau^2 = 265.71 \text{ MPa}$$

Outer inner surface displacement due to shrink fit
$$u_{inner} := \frac{D_{i,o} \cdot P_i}{2 \cdot E_o} \cdot \left(\frac{D_{i,o}^2 + D_{o,o}^2}{D_{o,o}^2 - D_{i,o}^2} + v_o \right) = 0.505 \text{ mm}$$

A.5 Geometry Preparation and Simplifications of The Downstream Flange for Simulation

The geometry of the downstream flange is simplified to ease meshing. Geometry can be simplified as long as the overall stiffness of the model is maintained. Examples of geometry simplification with meshing are presented below.

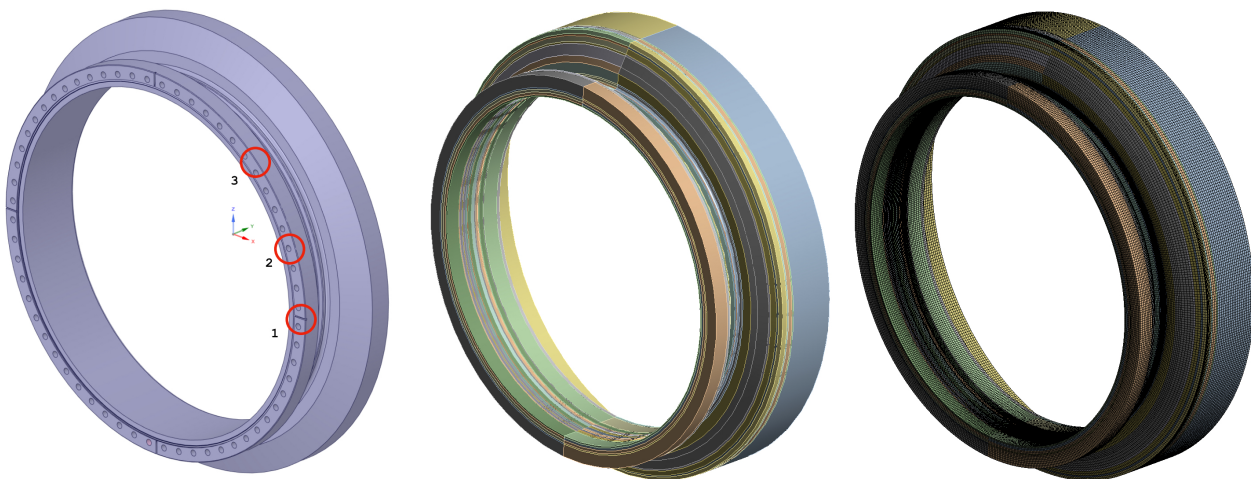


Figure A.10: Isometric view of the downstream flange. Left: original geometry. Center: Geometry after simplification. Right: after meshing

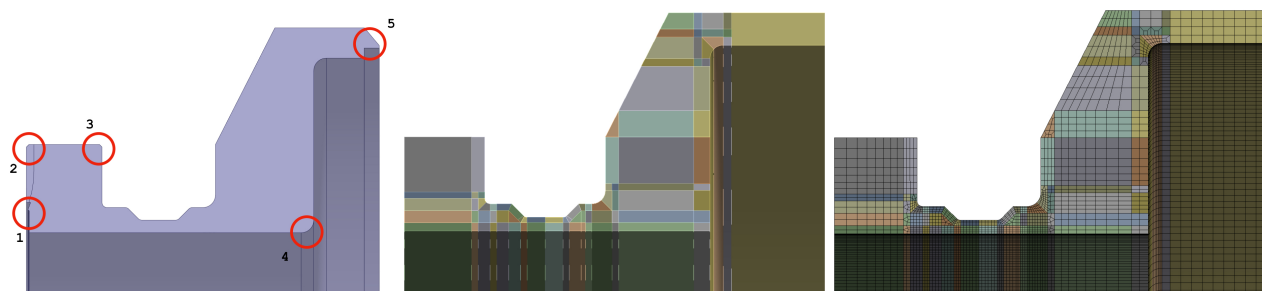


Figure A.11: Cross section view of the downstream flange. Left: original geometry. Center: Geometry after simplification. Right: after meshing

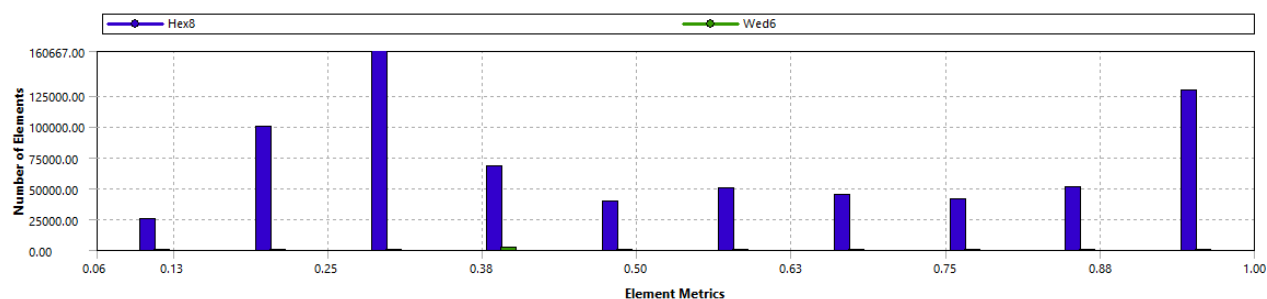


Figure A.12: Mesh metrics of the downstream flange showing element quality

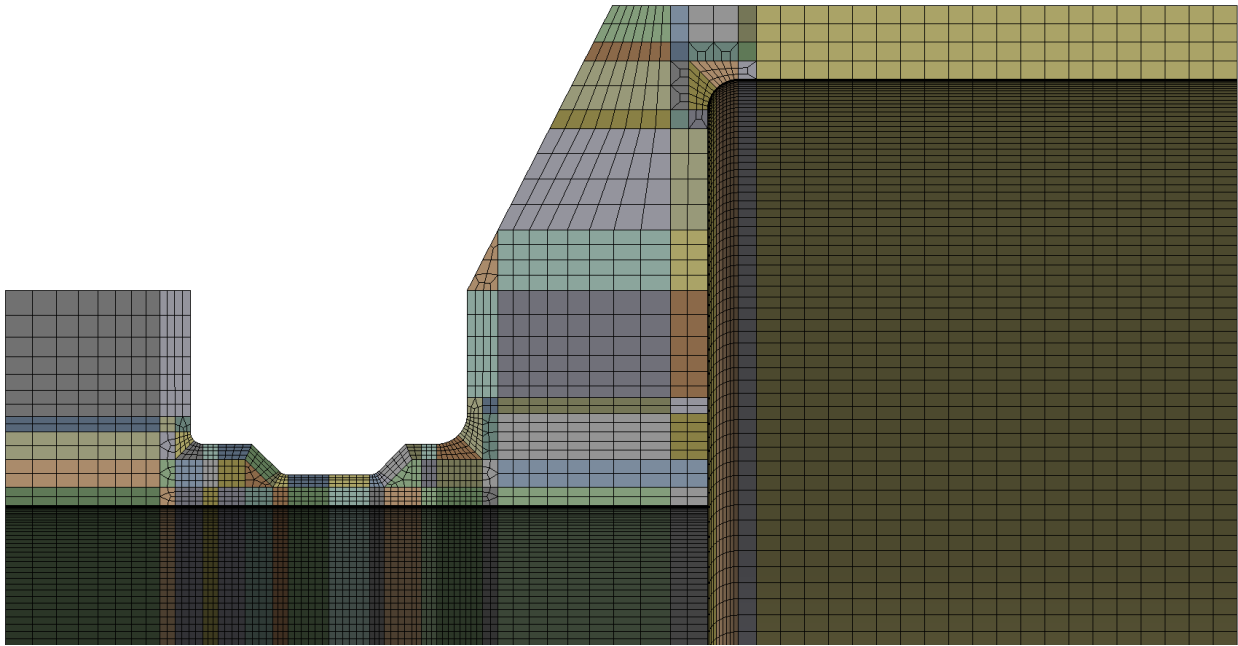
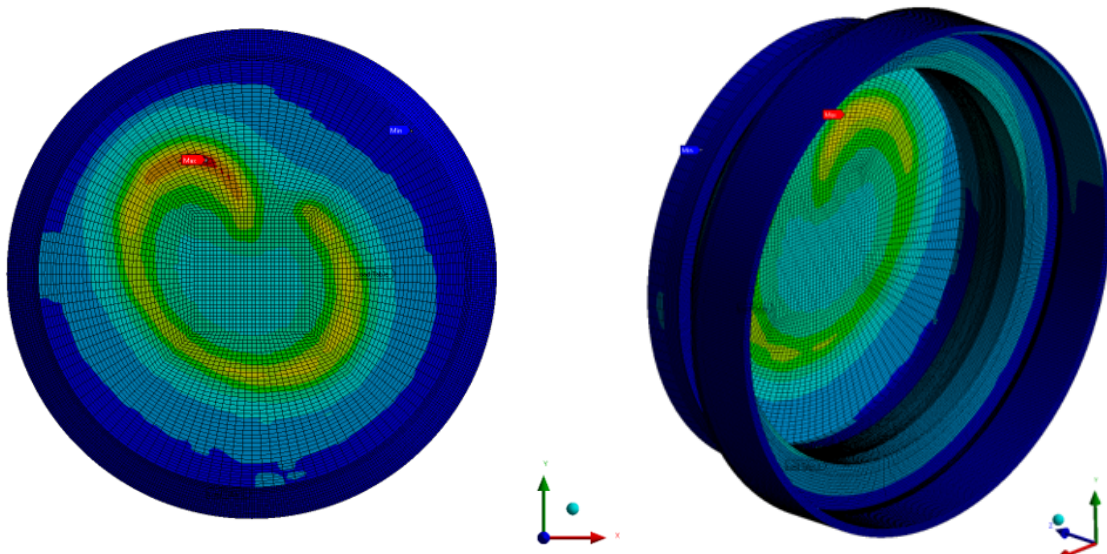


Figure A.13: Cross section view of the downstream flange after meshing



(a) Implementation of energy deposition from FLUKA file.

(b) Isometric view

Figure A.14: Implementation of energy deposition from FLUKA file. Max. temperature is 133 degrees Celsius. Min. temperature is 47 degrees Celsius.

A.6 Eurocode 3 Tables for Static Verification

Tables from Eurocode 3 (EN 1993-1-8) used to perform static verification of circumferential welds in the TDEs. Tables obtained from [5]

Standard and steel grade			Correlation factor β_w
EN 10025	EN 10210	EN 10219	
S 235 S 235 W	S 235 H	S 235 H	0,8
S 275 S 275 N/NL S 275 M/ML	S 275 H S 275 NH/NLH	S 275 H S 275 NH/NLH S 275 MH/MLH	0,85
S 355 S 355 N/NL S 355 M/ML S 355 W	S 355 H S 355 NH/NLH	S 355 H S 355 NH/NLH S 355 MH/MLH	0,9
S 420 N/NL S 420 M/ML		S 420 MH/MLH	1,0
S 460 N/NL S 460 M/ML S 460 Q/QL/QL1	S 460 NH/NLH	S 460 NH/NLH S 460 MH/MLH	1,0

Figure A.15: Correlation factor for fillet welds.

Resistance of members and cross-sections	γ_{M0} , γ_{M1} and γ_{M2} see EN 1993-1-1
Resistance of bolts	γ_{M2}
Resistance of rivets	
Resistance of pins	
Resistance of welds	
Resistance of plates in bearing	
Slip resistance - at ultimate limit state (Category C) - at serviceability limit state (Category B)	γ_{M3} $\gamma_{M3,scr}$
Bearing resistance of an injection bolt	γ_{M4}
Resistance of joints in hollow section lattice girder	γ_{M5}
Resistance of pins at serviceability limit state	$\gamma_{M6,scr}$
Preload of high strength bolts	γ_{M7}
Resistance of concrete	γ_c see EN 1992

NOTE: Numerical values for γ_M may be defined in the National Annex. Recommended values are as follows: $\gamma_{M2} = 1,25$; $\gamma_{M3} = 1,25$ and $\gamma_{M3,scr} = 1,1$; $\gamma_{M4} = 1,0$; $\gamma_{M5} = 1,0$; $\gamma_{M6,scr} = 1,0$; $\gamma_{M7} = 1,1$.

Figure A.16: Partial safety factor for welds.

A.7 Python Script for Eurocode 3 Static Verification

Python script used to post-process simulation data of the Target Dump External beam impact simulation. Using the Lasso library the script reads the history files written by LS-Dyna containing the simulation results. With in-built logic, the script filters the data based on body and coordinates allowing for the selection of elements in the weld locations. Based on the x - and y coordinates of the elements the stress tensors are rotated from a cartesian coordinate system into a cylindrical coordinate system compliant with the orientation defined in Eurocode 3. The Eurocode criteria are built into the script yielding a result for every element. The script outputs a data frame containing the assessment of all elements along with the maximum value for every criterion in every weld.

```

from lasso.dyna import D3plot, ArrayType, FilterType
import pandas as pd
import numpy as np
import plotly.express as px
import warnings
from pandas.core.common import SettingWithCopyWarning
from Functions import safety_timesteps, angle_calculate, matrix_rotation_z, eurocode_criteria_check,
↳ eurocode_boundary_criteria_1_func, eurocode_boundary_criteria_2_func, get_centroids,
↳ fatigue_normal_stress

# USER INPUT
c = 3272                                # Speed of sound in material in m/s
timestep_LSDyna = 0.00015                # Timestep in LS Dyna export file in s
total_timesteps = 502                   # Total number of timesteps in LS Dyna
radial = 0.012                           # Radial dimension (in beam dump case thickness of vessel) in m
longitudinal = 8                         # longitudinal dimension (in beam dump case length of vessel) in m
timestep_safety_factor = 20              # Script calculates needed timesteps based on Courant-Fredrichs-Lewy
↳ condition - to be sure this number of timesteps is increased by a safety factor to make sure
↳ everything is captured

x_range = [-1, 1]                        # x range for wanted elements (stress tensors)
y_range = [-1, 1]                        # y range for wanted elements (stress tensors)
z_range = [[1.227, 4.718]]               # z range for wanted elements (stress tensors)

# Circumferential settings
use_weld_locations = False               # Instead of defining z-interval, base the selection on weld
↳ location along z-axis

#weld_locations_z = [8.358]
weld_locations_z = [0.141, 0.499, 1.227, 2.100, 2.973, 3.845, 4.718, 5.446, 6.174, 6.902, 7.630,
↳ 8.358]
interval = 0.02                          # Search interval on each side of the weld in z-direction

# Longitudinal settings
longitudinal_mode = False                 # if set to true changes scaling of axis in plots
longitudinal_radial_selection = False
longitudinal_angles = [0, 45, 90, 135, 180, 225, 270, 315]
interval_angles = 2

part_lower = 83                          # Part IDs using to filter data to minimize data processed (LS Dyna
↳ part numbers)
part_upper = 244

f_u = 660000000                          # Nominel ultimate tensile strength of the weaker part joined (Pa)
beta_w = 1                               # Appropriate correlation factor taken from Table 4.1 in Eurocode
y_m2 = 1.25                             # Partial safety factor for fillet weld from table 4.1 in Eurocode

```

CHAPTER A. APPENDIX

```

include_initial_stress = False      # Decide whether to include or exclude initial stress (stress
↳ from timestep 0)
full_penetration = True            # If full penetration is false the criterias will be multiplied by
↳ a factor
partial_penetration_factor = 1.45  # Partial penetration factor

open_plots = False                 # Whether to print plots or not

# 3D premade plots
fig_geo = False                    # Shows all elements read from the d3plot in 3d
fig_selection = False              # Shows elements in selection based on user input
↳ in 3d
fig_selection_truescale = False     # Shows elements in selection based on user input
↳ but with same axis scale in 3d
fig_animation_euro_1 = True         # Shows eurocode 1 criteria in 3d of selected
↳ elements as a function of time
fig_animation_euro_2 = False        # Shows eurocode 2 criteria in 3d of selected
↳ elements as a function of time
fig_euro_1_max_timestep = False     # Shows elements inside selection from timestep
↳ where eurocode criteria 1 is maximum in 3d
fig_euro_1_max_timestep_normalised = False # Shows elements inside selection from timestep
↳ where eurocode criteria 1 is maximum and normalised with respect to criteria 1 boundary in 3d
fig_euro_2_max_timestep = False     # Shows elements inside selection from timestep
↳ where eurocode criteria 2 is maximum in 3d
fig_critical_elements_extract = False # Shows over simulation time if any element is
↳ above criteria 1
fig_stress_range = False            # Shows maximum element stress range for normal
↳ and shear stresses over simulation time

# 2D premade plots
fig_max_euro_1_welds_over_time = False # Shows the element with maximum stress for
↳ every timestep in every weld as a function of time
fig_euro_1_max_element_overtime = False # Shows variation in element that contains the
↳ global maximum euro 1 as a function of time
# USER INPUT END

warnings.simplefilter(action="ignore", category=SettingWithCopyWarning)
warnings.simplefilter(action='ignore', category=FutureWarning)
pd.set_option('display.max_rows', 200)
pd.set_option('display.max_columns', 500)
pd.set_option('display.width', 1000)

# Nth timestep data needed in LS Dyna export file in order to comply with Courant Lewy condition
safety_timesteps = safety_timesteps(radial, longitudinal, c, total_timesteps, timestep_LSDyna,
↳ timestep_safety_factor)
print(str(safety_timesteps) + " timesteps needs to be extracted")

# Creating list of states (timesteps) to extract from d3plot (reducing data pulled in memory
↳ dramatically)
timesteps_to_extract = np.linspace(0, total_timesteps, safety_timesteps)
timesteps_to_extract_rounded = sorted(set([round(num) for num in timesteps_to_extract]))
timesteps_to_extract_rounded[-1] = timesteps_to_extract_rounded[-1]-1
print("Data from these timesteps are extracted: " + str(timesteps_to_extract_rounded))

# Numpy list of parts needed for weld examination)
part_ids = np.array(list(range(part_lower, part_upper+1)))

# Reading the data from d3plot extract from LS Dyna
d3plot = D3plot("D:/Weld simulations/Global model/beam impact results updated/d3plot",
↳ state_array_filter=[ArrayType.global_timesteps, ArrayType.element_solid_stress,
↳ ArrayType.node_temperature], state_filter=timesteps_to_extract_rounded, buffered_reading=True)

```

CHAPTER A. APPENDIX

```

#d3plot = D3plot("D:\WeldProgram\Run 2 Jorge/d3plot", state_array_filter=[ArrayType.global_timesteps,
↪ ArrayType.element_solid_stress, ArrayType.node_temperature],
↪ state_filter=timesteps_to_extract_rounded, buffered_reading=True)
print("\n" + str(d3plot.arrays.keys()) + "\n")

# Extracting data based on keys that is going to be used

# Geometry based data
element_ids = d3plot.arrays[ArrayType.element_solid_ids]
element_nodes_index = d3plot.arrays[ArrayType.element_solid_node_indexes]
node_coordinates = d3plot.arrays[ArrayType.node_coordinates]

# State based data
timesteps = d3plot.arrays[ArrayType.global_timesteps]
element_solid_stress = d3plot.arrays[ArrayType.element_solid_stress]
print("Data from the following timesteps are extracted:\n " + str(timesteps) + "\n")

# Mask used for filtering the data so that only data from the chosen part is used
mask_elements = d3plot.get_part_filter(FilterType.SOLID, [part_ids])

# Applying filter to data (in selection)
element_ids_in_parts = element_ids[mask_elements]
element_nodes_index_in_parts = element_nodes_index[mask_elements]
element_solid_stress_in_parts = element_solid_stress[:, mask_elements, :, :]

# Some elements have duplicate node indexes - calculate centroid
element_centroids = get_centroids(element_nodes_index_in_parts, node_coordinates)

# Joins element ID column with element centroids
elements_coordinates = np.column_stack(
    (element_ids_in_parts, element_centroids))

# If the user has set use weld locations to True the following if statement creates a list of
↪ z-ranges for the script to look for elements
if use_weld_locations == True:
    z_range = []
    for i in weld_locations_z:
        lower_boundary = round(i + interval, 3)
        upper_boundary = round(i - interval, 3)
        boundary = [upper_boundary, lower_boundary]
        z_range.append(boundary)
    print("Intervals in z direction for welds: " + str(z_range))

# If the user has set longitudinal mode and radial selection to true, the following code creates the
↪ boundary conditions for element selection
if longitudinal_mode == True and longitudinal_radial_selection == True:
    angle_intervals = []
    for i in longitudinal_angles:
        lower_boundary = round(i + interval_angles, 3)
        upper_boundary = round(i - interval_angles, 3)
        boundary = [upper_boundary, lower_boundary]
        angle_intervals.append(boundary)
    print("Angle intervals for welds: " + str(angle_intervals))

# Calculates angle in XY plane for elements inside selection and appends result to array
angle = np.array([angle_calculate(row[1], row[2]) for row in elements_coordinates][:, None])
elements_coordinates = np.append(elements_coordinates, angle, axis=1)

# Extract the correct elements inside the boundary conditions (x, y, z and angle)
elements_in_selection = np.array([[0, 0, 0, 0]])
for angle in angle_intervals:

```

```

elements_in_weld = elements_coordinates[(elements_coordinates[:, 3] >= z_range[0][0]) &
↪ (elements_coordinates[:, 3] <= z_range[0][1]) &
    (elements_coordinates[:, 2] >= y_range[0]) &
↪ (elements_coordinates[:, 2] <= y_range[1]) &
    (elements_coordinates[:, 1] >= x_range[0]) &
↪ (elements_coordinates[:, 1] <= x_range[1]) &
    (elements_coordinates[:, 4] >= angle[0]) &
↪ (elements_coordinates[:, 4] <= angle[1])]

if elements_in_weld.shape[0] == 0:
    print("Did not find any elements in the angle interval " + str(angle) + "\n")
    elements_in_selection = np.append(elements_in_selection, elements_in_weld, axis=0)
elements_in_selection = np.delete(elements_in_selection, 0, 0)
elements_in_selection_for_stresslookup = elements_in_selection

else:

    # Extracts elements with centroid coordinate inside the given range in user input on all 3 axis
    elements_in_selection = np.array([[0, 0, 0, 0]])
    for loc in z_range:
        elements_in_weld = elements_coordinates[(elements_coordinates[:, 3] >= loc[0]) &
↪ (elements_coordinates[:, 3] <= loc[1]) &
            (elements_coordinates[:, 2] >= y_range[0]) &
↪ (elements_coordinates[:, 2] <= y_range[1]) &
            (elements_coordinates[:, 1] >= x_range[0]) &
↪ (elements_coordinates[:, 1] <= x_range[1])]

        if elements_in_weld.shape[0] == 0:
            print("Did not find any elements in the z-interval " + str(loc) + "\n")
            elements_in_selection = np.append(elements_in_selection, elements_in_weld, axis=0)
        elements_in_selection = np.delete(elements_in_selection, 0, 0)
        elements_in_selection_for_stresslookup = elements_in_selection

    # Calculates angle in XY plane for elements inside selection and appends result to array
    angle = np.array([angle_calculate(row[1], row[2]) for row in elements_in_selection])[:, None]
    elements_in_selection = np.append(elements_in_selection, angle, axis=1)

    # Repeat the data timestep times in order to migrate with stress tensors
    elements_in_selection = np.repeat(elements_in_selection, safety_timesteps, axis=0)

    # Tiles timesteps to fit elements in selection array and add the time to that array
    timesteps_corrected = np.expand_dims(np.tile(timesteps, int(elements_in_selection.shape[0] /
↪ timesteps.shape[0])), axis=1)
    elements_in_selection = np.append(elements_in_selection, timesteps_corrected, axis=1)

    # Find element index in element id array to look up stress tensors based on index
    element_index = np.expand_dims(np.searchsorted(element_ids, elements_in_selection_for_stresslookup[:,
↪ 0]), axis=1)
    elements_in_selection_for_stresslookup = np.append(elements_in_selection_for_stresslookup,
↪ element_index, axis=1)
    element_index_to_append = np.expand_dims(np.repeat(element_index, safety_timesteps), axis=1)
    elements_in_selection = np.append(elements_in_selection, element_index_to_append, axis=1)

    # Extracts element stress tensors based on ID from stress tensor array and appends to array
    stress_tensors_extracted = element_solid_stress[:, element_index, :, :]
    stress_tensors_to_append =
↪ stress_tensors_extracted.reshape(safety_timesteps*int(elements_in_selection.shape[0]/safety_timesteps),
↪ 6)
    stress_tensors = np.hstack(stress_tensors_extracted).reshape(-1, 6)
    stress_tensors = stress_tensors[:, [0, 3, 5, 1, 4, 2]]
    elements_in_selection = np.append(elements_in_selection, stress_tensors, axis=1)

```

CHAPTER A. APPENDIX

```
# The following if statement subtracts the initial stress (from timestep 0) from the stress tensors
↪ (for example if shrink fitting stresses dont want to be taken into account)
if include_initial_stress == False:
    a = []
    for row in elements_in_selection:
        if row[5] == 0:
            q = row[-6:]
            a.append(q)
        else:
            row[-6:] -= a[-1]

    for row in elements_in_selection:
        if row[5] == 0:
            row[-6:] = row[-6:] * 0

# Applies the matrix rotation matrix to all elements at all times and appends them to elements in
↪ selection
rotated_tensors = np.array([matrix_rotation_z(np.array(row[-6:])), row[4]] for row in
↪ elements_in_selection])
rotated_tensors = rotated_tensors.reshape(rotated_tensors.shape[0], 9)
rotated_tensors = rotated_tensors[:, [0, 1, 2, 4, 5, 8]]
elements_in_selection = np.append(elements_in_selection, rotated_tensors, axis=1)

# Calculates the eurocode boundary criteria
eurocode_criteria_1_boundary = eurocode_boundary_criteria_1_func(f_u, beta_w, y_m2)
eurocode_criteria_2_boundary = eurocode_boundary_criteria_2_func(f_u, y_m2)

# Calculates the eurocode criteria for all elements at all timesteps and append the results to
↪ elements in selection array

if longitudinal_mode == True: # If longitudinal mode is True different stress components of the
↪ stress tensor are used for the eurocode calculations.
    eurocode_criteria_1 = np.expand_dims(np.array([eurocode_criteria_check(row[16], row[14], row[17])
    ↪ for row in elements_in_selection]), axis=1)
    eurocode_criteria_2 = np.array(elements_in_selection[:, [16]])
else:
    eurocode_criteria_1 = np.expand_dims(np.array([eurocode_criteria_check(row[18], row[15], row[17])
    ↪ for row in elements_in_selection]), axis=1)
    eurocode_criteria_2 = np.array(elements_in_selection[:, [18]])

if full_penetration == False: # If full penetration is set to false the script multiplies the
↪ criterias by a user defined factor (partial penetration factor, based on coress section
↪ reduction)
    eurocode_criteria_1 = eurocode_criteria_1 * partial_penetration_factor
    eurocode_criteria_2 = eurocode_criteria_2 * partial_penetration_factor

# Appends results to array
elements_in_selection = np.append(elements_in_selection, eurocode_criteria_1, axis=1)
elements_in_selection = np.append(elements_in_selection, eurocode_criteria_2, axis=1)

# Calculates the normal stress as referred in the Eurocode of fatigue. Different inputs whether
↪ longitudinal or circumferential mode is selected
if longitudinal_mode == True:
    normal_stress = np.expand_dims(np.array([fatigue_normal_stress(row[16], row[14]) for row in
    ↪ elements_in_selection]), axis=1)
else:
    normal_stress = np.expand_dims(np.array([fatigue_normal_stress(row[18], row[15]) for row in
    ↪ elements_in_selection]), axis=1)
elements_in_selection = np.append(elements_in_selection, normal_stress, axis=1)

# Applies the partial penetration factor to the calculated normal stress as well
```

CHAPTER A. APPENDIX

```
if full_penetration == False:
    elements_in_selection[:, 21] = elements_in_selection[:, 21] * partial_penetration_factor

# Dataframe with all elements for making sure correct geometry is imported correctly
df_geo_plot = pd.DataFrame(columns=["Element ID", 'X', 'Y', 'Z'])
df_geo_plot["Element ID"] = elements_coordinates[:, 0]
df_geo_plot["X"] = elements_coordinates[:, 1]
df_geo_plot["Y"] = elements_coordinates[:, 2]
df_geo_plot["Z"] = elements_coordinates[:, 3]

# Dataframe with resulting data
df = pd.DataFrame(columns=["Element ID", 'X', 'Y', 'Z'])
df["Element ID"] = elements_in_selection[:, 0]
df["X"] = elements_in_selection[:, 1]
df["Y"] = elements_in_selection[:, 2]
df["Z"] = elements_in_selection[:, 3]
df["angle"] = elements_in_selection[:, 4]
df["time"] = elements_in_selection[:, 5]
df["Element index"] = elements_in_selection[:, 6]
df["x-stress"] = elements_in_selection[:, 7]
df["xy-stress"] = elements_in_selection[:, 8]
df["xz-stress"] = elements_in_selection[:, 9]
df["y-stress"] = elements_in_selection[:, 10]
df["yz-stress"] = elements_in_selection[:, 11]
df["z-stress"] = elements_in_selection[:, 12]
df["rr-stress"] = elements_in_selection[:, 13]
df["rf-stress"] = elements_in_selection[:, 14]
df["rz-stress"] = elements_in_selection[:, 15]
df["ff-stress"] = elements_in_selection[:, 16]
df["fz-stress"] = elements_in_selection[:, 17]
df["zz-stress"] = elements_in_selection[:, 18]
df["Euro. 1"] = elements_in_selection[:, 19]
df["Euro. 2"] = elements_in_selection[:, 20]
df["Normal-stress"] = elements_in_selection[:, 21]

# From here it's all plotting

# Calculate data-based values for axis-scales for plotting properly (true scale)
minimum_z = min(df_geo_plot["Z"])
maximum_z = max(df_geo_plot["Z"])
interval_z = maximum_z - minimum_z
scaling = interval_z / 2

# Minimum and maximum for Eurocode criteria 1 in MPa
min_euro_1_plot = min(df["Euro. 1"])
min_euro_1 = round(min_euro_1_plot/1000000, 1)
min_eu_1_check_row = df.iloc[[df["Euro. 1"].idxmin()]] # Takes the row in the dataframe
↳ containing the element that over all timesteps had the minimum result for eurocode criteria 1
max_euro_1_plot = max(df["Euro. 1"])
max_euro_1 = round(max_euro_1_plot/1000000, 1)
max_eu_1_check_row = df.iloc[[df["Euro. 1"].idxmax()]] # Takes the row in the dataframe
↳ containing the element that over all timesteps had the maximum result for eurocode criteria 1

# Minimum and maximum calculated for Eurocode criteria 2 in MPa
min_euro_2_plot = min(df["Euro. 2"])
min_euro_2 = round(min_euro_2_plot/1000000, 1)
min_eu_2_check_row = df.iloc[[df["Euro. 2"].idxmin()]] # Takes the row in the dataframe
↳ containing the element that over all timesteps had the minimum result for eurocode criteria 2
max_euro_2_plot = max(df["Euro. 2"])
max_euro_2 = round(max_euro_2_plot/1000000, 1)
```

CHAPTER A. APPENDIX

```

max_eu_2_check_row = df.iloc[[df["Euro. 2"].idxmax()]]      # Takes the row in the dataframe
↳ containing the element that over all timesteps had the maximum result for eurocode criteria 2

# Prints min, max and boundary criteria for eurocode criteria 1 and 2
print("\nEurocode criteria 1 MIN: " + str(min_euro_1) + " MPa - and MAX: " + str(max_euro_1) + "
↳ MPa")
print("Eurocode criteria 2 MIN: " + str(min_euro_2) + " MPa - and MAX: " + str(max_euro_2) + " MPa")
print("Eurocode boundary criteria 1 is: " + str(eurocode_criteria_1_boundary/1000000) + " MPa")
print("Eurocode boundary criteria 2 is: " + str(eurocode_criteria_2_boundary/1000000) + " MPa\n")

# Prints preview of dataframe containing results
print(df)
print("\n")

# Plotting based on user choices - if open_plots is set to True the script will print any plot that
↳ is set to True
# Description of the different plots can be found in the input section. All plots are based on
↳ extraction in the dataframe
if open_plots == True:

    if fig_geo == True:
        figgeo = px.scatter_3d(df_geo_plot, x='X', y='Y', z='Z', title="Geometry imported",
↳ labels={"z": "z - positive direction is beam direction"}, )
        figgeo.update_layout(scene=dict(xaxis=dict(range=[-scaling, scaling], ),
↳ yaxis=dict(range=[-scaling, scaling], ), zaxis=dict(range=[min(df_geo_plot["Z"]),
↳ max(df_geo_plot["Z"])]), ), ))
        figgeo.show()

    if fig_selection == True:
        figselection = px.scatter_3d(df, x='X', y='Y', z='Z', title="Selected elements", labels={"z":
↳ "z - positive direction is beam direction"})
        if longitudinal_mode == True:
            figselection.update_layout(scene=dict(xaxis=dict(range=[-scaling, scaling], ),
↳ yaxis=dict(range=[-scaling, scaling], ), zaxis=dict(range=[min(df_geo_plot["Z"]),
↳ max(df_geo_plot["Z"])]), ), ))
            figselection.show()

    if fig_selection_truescale == True:
        figselectiontrue = px.scatter_3d(df, x='X', y='Y', z='Z', title="Selected elements, axis with
↳ same scaling", labels={"z": "z - positive direction is beam direction"})
        figselectiontrue.update_layout(scene=dict(xaxis=dict(range=[-scaling, scaling], ),
↳ yaxis=dict(range=[-scaling, scaling], ), zaxis=dict(range=[min(df_geo_plot["Z"]),
↳ max(df_geo_plot["Z"])]), ), ))
        figselectiontrue.show()

    if fig_animation_euro_1 == True:
        figanimationeuro1 = px.scatter_3d(df, x='X', y='Y', z='Z', animation_frame="time",
↳ animation_group="Euro. 1", color="Euro. 1", range_color=[min_euro_1_plot,
↳ max_euro_1_plot], title="Selected elements of weld and EuroCode criteria 1", labels={"z":
↳ "z - positive direction is beam direction"})
        if longitudinal_mode == True:
            figanimationeuro1.update_layout(scene=dict(xaxis=dict(range=[-scaling, scaling], ),
↳ yaxis=dict(range=[-scaling, scaling], ), zaxis=dict(range=[min(df_geo_plot["Z"]),
↳ max(df_geo_plot["Z"])]), ), ))
            figanimationeuro1.show()

    if fig_animation_euro_2 == True:
        figanimationeuro2 = px.scatter_3d(df, x='X', y='Y', z='Z', animation_frame="time",
↳ animation_group="Euro. 2", color="Euro. 2", range_color=[min_euro_2_plot,
↳ max_euro_2_plot], title="Selected elements of weld and EuroCode criteria 2", labels={"z":
↳ "z - positive direction is beam direction"})

```



```

if longitudinal_mode == True:
    figanimationeuro2.update_layout(scene=dict(xaxis=dict(range=[-scaling, scaling], ),
    ↪ yaxis=dict(range=[-scaling, scaling], ), zaxis=dict(range=[min(df_geo_plot["Z"]),
    ↪ max(df_geo_plot["Z"])]), ), ))
figanimationeuro2.show()

if fig_euro_1_max_timestep == True:
    # Extraction and plotting of 3D Plot of Eurocode 1 results at timestep where maximum value is
    ↪ achieved
    df_max_timestep = df.loc[df['time'] == max_eu_1_check_row.iloc[0]["time"]] # Extracts
    ↪ elements from dataframe where timestep is equal to timestep where maximum eurocode 1 was
    ↪ found
    max_euro1_timestep_minimum = min(df_max_timestep["Euro. 1"])
    max_euro1_timestep_maximum = max(df_max_timestep["Euro. 1"])
    figeuro1maxtime = px.scatter_3d(df_max_timestep, x='X', y='Y', z='Z', color="Euro. 1",
    ↪ title="Selected elements and EuroCode criteria 1 in maximum timestep: " +
    ↪ str(round(max_eu_1_check_row.iloc[0]["time"], 5)) + " s. Max. is " +
    ↪ str(round(max_euro1_timestep_maximum / 1000000, 1)) + " MPa. Min. is " +
    ↪ str(round(max_euro1_timestep_minimum / 1000000, 1)) + " MPa", labels={"Z": "z - positive
    ↪ direction is beam direction"})
    if longitudinal_mode == True:
        figeuro1maxtime.update_layout(scene=dict(xaxis=dict(range=[-scaling, scaling], ),
        ↪ yaxis=dict(range=[-scaling, scaling], ), zaxis=dict(range=[min(df_geo_plot["Z"]),
        ↪ max(df_geo_plot["Z"])]), ), ))
    figeuro1maxtime.show()

if fig_euro_1_max_timestep_normalised == True:
    # Extraction and plotting of 3D Plot of Eurocode 1 results at timestep where maximum value is
    ↪ achieved - normalised
    df_max_timestep["Euro. 1"] = (df_max_timestep["Euro. 1"] / eurocode_criteria_1_boundary) #
    ↪ Divides eurocode criteria 1 with boundary value on all elements at all timesteps
    max_euro1_timestep_normalised_minimum = min(df_max_timestep["Euro. 1"])
    max_euro1_timestep_normalised_maximum = max(df_max_timestep["Euro. 1"])
    figeuro1maxtimenormalised = px.scatter_3d(df_max_timestep, x='X', y='Y', z='Z', color="Euro.
    ↪ 1", title="Euro. crit. 1 max. time: " + str(round(max_eu_1_check_row.iloc[0]["time"], 5))
    ↪ + " s and normalised to Euro. crit. 1: " + str(round(eurocode_criteria_1_boundary /
    ↪ 1000000, 1)) + " MPa. Max. is " + str(round(max_euro1_timestep_normalised_maximum, 4)) +
    ↪ ". Min. is " + str(round(max_euro1_timestep_normalised_minimum, 4)), labels={"Z": "z -
    ↪ positive direction is beam direction"})
    if longitudinal_mode == True:
        figeuro1maxtimenormalised.update_layout(scene=dict(xaxis=dict(range=[-scaling, scaling],
        ↪ ), yaxis=dict(range=[-scaling, scaling], ), zaxis=dict(range=[min(df_geo_plot["Z"]),
        ↪ max(df_geo_plot["Z"])]), ), ))
    figeuro1maxtimenormalised.show()

if fig_euro_2_max_timestep == True:
    # Extraction and plotting of 3D Plot of Eurocode 2 results at timestep where maximum value is
    ↪ achieved
    max_euro2_timestep_minimum = min(df_max_timestep["Euro. 2"])
    max_euro2_timestep_maximum = max(df_max_timestep["Euro. 2"])
    figeuro2maxtime = px.scatter_3d(df_max_timestep, x='X', y='Y', z='Z', color="Euro. 2",
    ↪ range_color=[max_euro2_timestep_minimum, max_euro2_timestep_maximum], title="Euro. crit.
    ↪ 2 max. at: " + str(round(max_eu_2_check_row.iloc[0]["time"], 5)) + " seconds. Max. is " +
    ↪ str(round(max_euro2_timestep_maximum / 1000000, 1)) + " MPa. Min. is " +
    ↪ str(round(max_euro2_timestep_minimum / 1000000, 1)) + " Mpa", labels={"Z": "z - positive
    ↪ direction is beam direction"})
    if longitudinal_mode == True:
        figeuro2maxtime.update_layout(scene=dict(xaxis=dict(range=[-scaling, scaling], ),
        ↪ yaxis=dict(range=[-scaling, scaling], ), zaxis=dict(range=[min(df_geo_plot["Z"]),
        ↪ max(df_geo_plot["Z"])]), ), ))
    figeuro2maxtime.show()

```

```

if fig_euro_1_max_element_overtime == True:
    # Extraction and plotting of 2D plot of E1_criterion value over time
    df_element_max_euro1 = df.loc[df['Element ID'] == max_eu_1_check_row.iloc[0]["Element ID"]]
    ↪ # Extracts element from dataframe that have same element ID as the Element ID for the
    ↪ maximum eurocode criteria 1 row
    figeuro1maxelementovertime = px.line(df_element_max_euro1, x="time", y="Euro. 1",
    ↪ title="Element containing maximum eurocode criteria 1 over simulation time",
    ↪ labels={"time": "Time (seconds)", "Euro. 1": "Eurocode criteria 1 (Pa)"}
    figeuro1maxelementovertime.show()

if fig_max_euro_1_welds_over_time == True:
    data = pd.DataFrame(columns=["Element ID", 'X', 'Y', 'Z']) # Creates empty dataframe
    ↪ for later appending
    for elements in z_range:
        weldlocation = df[df["Z"] >= elements[0]] # Extracts all the
        ↪ elements from the dataframe that are above the lower boundary in the weld interval
        weldlocation = weldlocation[weldlocation["Z"] <= elements[1]] # Extracts all elements
        ↪ from the extracted dataframe that are below the upper boundary in the weld interval
        weldlocation["Weld"] = str(elements) # Appends weld interval
        ↪ to a new column in the dataframe
        for time in timesteps:
            timestep_splitting = weldlocation[weldlocation["time"] == time] # Extracts
            ↪ all elements that iteration timestep
            max_timestep = timestep_splitting.loc[[timestep_splitting["Euro. 1"].idxmax()]] #
            ↪ Finds the row that has maximum eurocode 1 criteria for that that weld at that
            ↪ timestep
            data = data.append(max_timestep, ignore_index=True) # Appends
            ↪ this row to the empty dataframe
        max_euro_1_each_timestep_each_weld = px.line(data, x="time", y="Euro. 1", color='Weld')
        max_euro_1_each_timestep_each_weld.show()

if fig_critical_elements_extract == True:
    critical_elements_extract = pd.DataFrame(columns=["Element ID", 'X', 'Y', 'Z']) # Creates
    ↪ empty dataframe for later appending
    critical_elements = df[df["Euro. 1"] > eurocode_criteria_1_boundary] # Finds
    ↪ all elements that are above the eurocode criteria 1
    critical_elements["Critical/Non-critical"] = "Critical" # Appends
    ↪ a column to the dataframe with a string "critical"
    non_critical_elements = df[df["Euro. 1"] <= eurocode_criteria_1_boundary] # Finds
    ↪ all elements that are below the eurocode criteria 1
    non_critical_elements["Critical/Non-critical"] = "Non-critical" # Appends
    ↪ a column to the dataframe with a string "Non-critical"
    critical_elements_extract = critical_elements_extract.append(critical_elements,
    ↪ ignore_index=True) # Appends both dataframes to the empty dataframe
    ↪ critical_elements_extract
    critical_elements_extract = critical_elements_extract.append(non_critical_elements,
    ↪ ignore_index=True)
    critical_elements_extract_plot = px.scatter_3d(critical_elements_extract, x='X', y='Y',
    ↪ z='Z', animation_frame="time", color="Critical/Non-critical", title="Critical elements in
    ↪ selected geometry (Euro. 1)", labels={"z": "z - positive direction is beam direction"})
    if longitudinal_mode == True:
        critical_elements_extract_plot.update_layout(scene=dict(xaxis=dict(range=[-scaling,
        ↪ scaling], ), yaxis=dict(range=[-scaling, scaling], ),
        ↪ zaxis=dict(range=[min(df_geo_plot["Z"]), max(df_geo_plot["Z"])]), ), ))
    critical_elements_extract_plot.show()

if fig_stress_range == True:
    start = 0
    end = len(df["Element ID"])
    steps = np.linspace(0, end, num=int(end / safety_timesteps))

```

```

stress_range = pd.DataFrame(columns=["Element ID"])
for i in steps: # for loop the iterates through each element' timesteps
    element = df.iloc[start:start + safety_timesteps]

    max_normal_stress = element["zz-stress"].max() # Finds maximum normal stress in the
    ↪ element
    min_normal_stress = element["zz-stress"].min() # Finds minimum normal stress in the
    ↪ element
    max_tau_par_stress = element["fz-stress"].max() # Finds maximum shear stress in the
    ↪ element
    min_tau_par_stress = element["fz-stress"].min() # Finds minimum shear stress in the
    ↪ element

    element_normal_stress_range = (max_normal_stress - min_normal_stress) # maximum normal
    ↪ stress range
    element_tau_par_stress_range = (max_tau_par_stress - min_tau_par_stress) # maximum shear
    ↪ stress range

    # creates dataframe containing x, y, z, element id, max normal stress range, max shear
    ↪ stress range
    append = element.iloc[0][["Element ID", "X", "Y", "Z"]]
    append["normal stress range"] = element_normal_stress_range
    append["tau par stress range"] = element_tau_par_stress_range
    stress_range = stress_range.append(append, ignore_index=True)

    start = start + safety_timesteps

# From new dataframe containing only maximum normal and shear stress ranges for every element
↪ over element time, extract data for the highest normal stress range element
max_normal_stress_range = stress_range.iloc[[stress_range["normal stress range"].idxmax()]]
crit_element_normal = int(max_normal_stress_range["Element ID"])
normal_stress_range_df = df[df["Element ID"] == crit_element_normal]
max_normal_stress_timestep = px.line(normal_stress_range_df, x="time", y="zz-stress",
↪ title="Normal stress", labels={"time": "Time (seconds)", "zz-stress": "Normal stress
↪ (zz-stress)"})

# From new dataframe containing only maximum normal and shear stress ranges for every element
↪ over element time, extract data for the highest shear stress range element
max_shear_stress_range = stress_range.iloc[[stress_range["tau par stress range"].idxmax()]]
crit_element_shear = int(max_shear_stress_range["Element ID"])
shear_stress_range_df = df[df["Element ID"] == crit_element_shear]
max_shear_stress_timestep = px.line(shear_stress_range_df, x="time", y="fz-stress",
↪ title="Shear stress", labels={"time": "Time (seconds)", "fz-stress": "Shear stress
↪ (fz-stress)"})

# 3D plot of highest stress range for all elements (time not the same for elements)
normal_stress_range_plot = px.scatter_3d(stress_range, x='X', y='Y', z='Z', color="normal
↪ stress range", title="Normal stress range (maximum and minimum over simulation time pr
↪ element)", labels={"z": "z - positive direction is beam direction", "normal stress
↪ range": "Normal stress range (zz-stress)"})
tau_par_stress_range_plot = px.scatter_3d(stress_range, x='X', y='Y', z='Z', color="tau par
↪ stress range", title="Shear stress range (maximum and minimum over simulation time pr
↪ element)", labels={"z": "z - positive direction is beam direction", "tau par stress
↪ range": "tau par stress range (fz-stress)"})

normal_stress_range_plot.show()
max_normal_stress_timestep.show()
tau_par_stress_range_plot.show()
max_shear_stress_timestep.show()

max_shear_stress_timestep.write_html("D:/max_shear_stress_timestep.html")

```

CHAPTER A. APPENDIX

```

normal_stress_range_df.to_csv("D:/normal_stress_range_weld_df.csv")
shear_stress_range_df.to_csv("D:/shear_stress_range_weld_df.csv")

import math
from math import atan, degrees
import numpy as np

def safety_timesteps(radial, longitudinal, c, total_timesteps, timesteps_LSDyna,
↳ timestep_safety_factor):
    # Time step determination
    timestep_radial = 2*(radial/c) # Needed timestep to capture
    ↳ radial values
    timestep_longitudinal = 2*(longitudinal/c) # Needed timestep to capture
    ↳ longitudinal values

    # Determines minimum nth timesteps needed based on user input (Courant-condition)
    if timestep_longitudinal >= timesteps_LSDyna and timestep_longitudinal <= timestep_radial:
        nth_timestep_needed = timestep_longitudinal / timesteps_LSDyna
    elif timestep_longitudinal >= timesteps_LSDyna and timestep_longitudinal >= timestep_radial:
        nth_timestep_needed = timestep_longitudinal / timesteps_LSDyna
    elif timestep_radial >= timesteps_LSDyna and timestep_radial <= timestep_longitudinal:
        nth_timestep_needed = timestep_radial / timesteps_LSDyna
    elif timestep_radial >= timesteps_LSDyna and timestep_radial >= timestep_longitudinal:
        nth_timestep_needed = timestep_radial / timesteps_LSDyna
    else:
        print("Warning - the timesteps in LS Dyna are not small enough to capture conditions in parts
        ↳ based on Courant-Fredrichs-Lewy condition")
    safety_timesteps = round(total_timesteps/(nth_timestep_needed/timestep_safety_factor)) # Nth
    ↳ timestep data needed in LS Dyna export file
    return safety_timesteps

# Calculates angle of coordinate in x, y plane from 0 to 360, x being 0
def angle_calculate(length_a, length_b):
    if length_a > 0 and length_b > 0:
        angle = (degrees(atan(length_b / length_a)))
    elif length_a < 0 and length_b > 0:
        angle = (degrees(atan(length_b / length_a))) + 180
    elif length_a < 0 and length_b < 0:
        angle = (degrees(atan(length_b / length_a))) + 180
    elif length_a > 0 and length_b < 0:
        angle = (degrees(atan(length_b / length_a))) + 360
    elif length_a > 0 and length_b == 0:
        angle = 0
    elif length_a == 0 and length_b > 0:
        angle = 90
    elif length_a < 0 and length_b == 0:
        angle = 180
    elif length_a == 0 and length_b < 0:
        angle = 270
    else:
        angle = "Error"
    return angle

# Turns every stress tensor array of shape 6 into a symemtric matrix and rotates it around z-axis
↳ based on the angle.
def matrix_rotation_z(matrix_array, angle):
    matrix = np.zeros((3,3))
    inds = np.triu_indices_from(matrix)
    matrix[inds] = matrix_array
    matrix[(inds[1], inds[0])] = matrix_array

```

```

rotation_z = np.array([[math.cos(math.radians(angle)), -math.sin(math.radians(angle)), 0],
↪ [math.sin(math.radians(angle)), math.cos(math.radians(angle)), 0], [0, 0, 1]])
rotation_z_transposed = np.transpose(rotation_z)
rotated_tensor = np.matmul(np.matmul(rotation_z_transposed, matrix), rotation_z)
return rotated_tensor

def matrix_rotation_f(matrix_array):
    angle = 45
    matrix = np.zeros((3, 3))
    inds = np.triu_indices_from(matrix)
    matrix[inds] = matrix_array
    matrix[(inds[1], inds[0])] = matrix_array
    rotation_f = np.array([[math.cos(math.radians(angle)), 0, -math.sin(math.radians(angle))], [0, 1,
↪ 0], [math.sin(math.radians(angle)), 0, math.cos(math.radians(angle))]])
    rotation_f_transposed = np.transpose(rotation_f)
    rotation_tensor = np.matmul(np.matmul(rotation_f, matrix), rotation_f_transposed)
    return rotation_tensor

# Calculates eurocode criteria 1
def eurocode_criteria_check(sigma_per, tau_per, tau_par):
    check = (sigma_per ** 2 + 3 * (tau_per ** 2 + tau_par ** 2)) ** 0.5
    return check

# Calculated the boundary criteria for Eurocode criteria 1
def eurocode_boundary_criteria_1_func(f_u_func, beta_w_func, y_m2_func):
    criteria = f_u_func / (beta_w_func * y_m2_func)
    return criteria

# Calculated the boundary criteria for Eurocode criteria 2
def eurocode_boundary_criteria_2_func(f_u_func, y_m2_func):
    criteria = 0.9 * f_u_func / y_m2_func
    return criteria

# Calculates nominal stress for fatigue examination
def fatigue_normal_stress(sigma_per, tau_per):
    normal_stress = (sigma_per ** 2 + tau_per ** 2) ** 0.5
    return normal_stress

# Calculates the centroid of the element using elements connectivity and node coordinated
def get_centroids(inds, coords):
    inds_sorted = np.sort(inds, axis=1)
    diff = np.diff(inds_sorted, axis=1)
    mask = np.ones(inds_sorted.shape, dtype=int)
    mask[:, 1:] = np.not_equal(diff, 0)
    mask = (mask / mask.sum(keepdims=True, axis=1))[:, :, None]
    centroids = (coords[inds_sorted] * mask).sum(axis=1)
    return centroids

```

A.8 APDL Script for Implementation of Boundary Displacements In 2D Axisymmetric Simulation of Weld

The APDL script creates two tables containing axial displacements as function of position through vessel thickness and time. This allows Ansys to interpolate in both time, position and displacement.

```

1  ! Two tables: one for upstream boundary and another for downstream boundary
2  ! Deletion of existing tables
3  *DEL,datatableupstream
4  *DEL,datatabledownstream
5
6  ! Creation of new tables. 1002 corresponds to the number of timesteps and 6 refers
7  ! to 6 datapoints through the thickness of the vessel with the header indication
8  ! the position of the data.
9  *DIM,datatableupstream,TABLE,1002,6,,Time,X
10 *DIM,datatabledownstream,TABLE,1002,6,,Time,X
11
12 ! Appending the content of the csv files extracted from LS-Dyna to the tables
13 *TREAD,datatabledownstream,D:/downstreamdisplacements,csv
14 *TREAD,datatableupstream,D:/downstreamdisplacements,csv
15
16 ! Applying the table data to the boundaries indication UY to be the displacements
17 ! in the Y direction (axial direction)
18 D,Downstream,UY,%datatabledownstream%
19 D,Upstream,UY,%datatableupstream%
```

The following figure is a snippet of the csv file fed to the APDL script when running the simulation. The header contains the radial coordinate referring to the axial displacements (position through the thickness of the vessel). Rows are the axial displacement at every timestep.

	A	B	C	D	E	F	G
1		0.349691	0.351599	0.353508	0.356234	0.35896	0.361686
2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
3	5.00E-05	-1.23E-06	-1.15E-06	-1.05E-06	-9.32E-07	-7.57E-07	-6.15E-07
4	1.00E-04	-9.14E-06	-9.24E-06	-9.36E-06	-9.44E-06	-9.53E-06	-9.67E-06
5	1.50E-04	-3.08E-05	-3.12E-05	-3.14E-05	-3.17E-05	-3.21E-05	-3.25E-05
6	2.00E-04	-5.24E-05	-5.28E-05	-5.31E-05	-5.36E-05	-5.40E-05	-5.45E-05
7	2.50E-04	-6.93E-05	-6.93E-05	-6.94E-05	-6.94E-05	-6.95E-05	-6.95E-05
8	3.00E-04	-8.58E-05	-8.56E-05	-8.56E-05	-8.54E-05	-8.53E-05	-8.53E-05
9	3.50E-04	-9.70E-05	-9.68E-05	-9.67E-05	-9.65E-05	-9.63E-05	-9.61E-05

Coordinate
Referring to
position through the
thickness of the
vessel

Time

Axial displacements

Figure A.17: Snippet of the csv file fed to the APDL script in 2D axisymmetric simulation.

A.9 Python Script for Estimation of Damping Ratio Using LDV Data

Python script used to post-process LDV data. First section opens, manipulates and filter the LDV data. Second section identifies the peaks of the filtered data which is used for fitting the exponential decaying function in third section. The FFT of the damped data is performed to find the damped frequency of the system.

```

1  # Libraries used
2  import numpy as np
3  import pandas as pd
4  from matplotlib import pyplot as plt
5  import math
6  from scipy.optimize import curve_fit
7
8  #Open, manipulate and filter LDV data.
9  Data = pd.read_csv("D:\Dampdata.csv")
10
11  Timeglobal = Data.iloc[19500:23000,0]
12  Velocityglobal = Data.iloc[19500:23000,1]
13
14  Time = Data.iloc[20051:22571,0]
15  Velocity = Data.iloc[20051:22571,1]
16
17  Time = Time.to_numpy()
18  Velocity = Velocity.to_numpy()
19  Velocityglobal = Velocityglobal.to_numpy()
20
21  Velocityglobal = Velocityglobal - Velocity.mean()
22  Velocity = Velocity - Velocity.mean()
23
24  #Identify peaks
25  peaks = []
26  n = 10
27  for i in range(n,len(Velocity)-n):
28      if max(Velocity[i-n:i+n]) == Velocity[i]:
29          peaks.append(i)
30
31  #Fitting of exponential function to peaks
32  def func(x, a, b):
33      return a*np.exp(b*x)
34
35  popt, pcov = curve_fit(func, Time[peaks], Velocity[peaks])
36  a = popt[0]
37  b = popt[1]
38
39  #Plotting
40  plt.plot(Time[peaks], Velocity[peaks], 'ro')
41  plt.plot(Time, Velocity)
42  plt.plot(Timeglobal, Velocityglobal, 'b')
43  plt.plot(Time[peaks], a*np.exp(b*Time[peaks]))
44  plt.title("Velocity of TDE upon beam impact")
45  plt.xlabel("Time [s]")
46  plt.ylabel("Velocity [m/s]")
47  plt.show()
48
49  print(a)
50  print(b)

```


CHAPTER A. APPENDIX

```
51
52 #FFT of damped LDV data
53 def compute_response(x, y):
54     N = len(y) # Number of samples
55     k = np.arange(N) # Frequency indices
56
57     Ts = x[N - 1] / N # Estimate sampling period from last time step
58     Fs = 1 / Ts # Sampling frequency
59     freq2side = (k * Fs) / N # Generate 2-sided frequency axis
60     freq = freq2side[range(int(N / 2))] # Discard negative frequencies
61
62     response = np.fft.fft(y) / N # Compute FFT
63     response = response[range(int(N / 2))] # Discard negative frequencies
64     response = np.abs(response) # Compute amplitude response
65
66     return response, freq
67
68 def plot_results(x, y, freq, response):
69     # Create figure and axes
70     fig, ax = plt.subplots(1, 2, figsize=(10, 4))
71     fig.tight_layout(pad=2.0)
72
73     # Time domain plot
74     ax[0].plot(x, y)
75     ax[0].set_xlabel("Time")
76     ax[0].set_ylabel("Velocity")
77
78     # Frequency response plot
79     ax[1].plot(freq, response, 'r-')
80     ax[1].set_xlabel('freq (Hz)')
81     ax[1].set_ylabel('|Y(freq)|')
82
83     return fig, ax
84
85 x = Time
86 y = Velocity
87
88 response, freq = compute_response(x, y)
89 freq_peak = freq[np.argmax(response)]
90 Y_peak = response[np.argmax(response)]
91 fig, ax = plot_results(x, y, freq, response)
92
93 #Add vertical dashed line at peak frequency
94 ax[1].axvline(freq_peak, 0, Y_peak, linestyle='--', color='k',
95              label=f"Peak frequency: {freq_peak:.1f} Hz")
96 ax[1].legend()
97 plt.show()
98
99 dampedfreq = Y_peak
```

Bibliography

- [1] How Much Do Airplanes Weigh? (With 20 Examples) – Pilot Teacher.
- [2] Metals - Latent Heat of Fusion.
- [3] Metals - Specific Heats.
- [4] The official CERN FLUKA website.
- [5] EN 1993-1-8: Eurocode 3: Design of steel structures - Part 1-8: Design of joints. Technical report, 2005.
- [6] Technical Data sheet: Duplex Stainless Steel (ATI 2205). 2015.
- [7] CERN annual report 2016. 2016.
- [8] Ahdida C., Bozzato D., Calzolari D., Cerutti F., Charitonidis N., Cimmino A., Coronetti A., D'Alessandro G. L., Donadon S., Esposito L. S., Froeschl R., García Alía R., Gerbershagen A., Gilardoni S., Horváth D., Hugo G., Infantino A., Kouskoura V., Lechner A., Lefebvre B., Lerner G., Magistris M., Manousos A., Moryc G., Ogallar Ruiz F., Pozzi F., Prelicpean D., Roesler S., Rossi R., Sabaté Gilarte M., Salvat Pujol F., Schoofs P., Stránský V., Theis C., Tsinganis A., Versaci R., Vlachoudis V., Waets A., and Widorski M. New Capabilities of the FLUKA Multi-Purpose Code. *Frontiers in Physics* 9, 788253 (2022), 2022.
- [9] Roger Bailey and Paul Collier. Standard Filling Schemes for Various LHC Operation Modes (Revised). Technical report, 2008.
- [10] Giuseppe Battistoni, Till Boehlen, Francesco Cerutti, Pik Wai Chin, Luigi Salvatore Esposito, Alberto Fassò, Alfredo Ferrari, Anton Lechner, Anton Empl, Andrea Mairani, Alessio Mereghetti, Pablo Garcia Ortega, Johannes Ranft, Stefan Roesler, Paola R. Sala, Vasilis Vlachoudis, and George Smirnov. Overview of the FLUKA code. *Annals of Nuclear Energy*, 82:10–18, 7 2015.
- [11] Battistoni G., Boehlen T., Cerutti F., Chin P.W., Esposito L.S., Fassò A., Ferrari A., Lechner A., Empl A., Mairani A., Mereghetti A., Garcia Ortega P., Ranft J., Roesler S., Sala P.R., Vlachoudis V., and Smirnov G. *Frontiers in Physics* 9. *Annals of Nuclear Energy* 82, pages 10–18, 2015.

CHAPTER BIBLIOGRAPHY

- [12] Bergengren Y., Larsson M., and Melander A. *Fatigue properties of stainless sheet steels in air at room temperature*, volume 11. Materials Science and Technology, 12 edition, 1994.
- [13] EN-MME. Welding Performance Qualification Record (WPQR) according to ISO15613 of Uranus 45. Technical report, 2022.
- [14] T Hakulinen, S Di Luca, G Godineau, R Nunes, G Smith, and Geneva Switzerland. Redesign and upgrade of the LHC Access Control System and system design. *Proceedings of IPAC2021*, pages 2249–2251, 2021.
- [15] Armin Hermann, Lanfranco Belloni, Gerhard John; Krige, Ulrike Mersits, and Dominique Pestre. *Launching the European Organization for Nuclear Research*. 1987.
- [16] Jacob. P and Goulding. L. *An Explicit Finite Element Primer*. 2002.
- [17] James L. Meriam, L. G. Kraige, and J. N. Bolton. *Engineering Mechanics: Dynamics*. 9th edition edition, 2018.
- [18] Michael Krause. *CERN how we found the Higgs boson*. World Scientific, 2014.
- [19] J Maestre, P Andreu-Muñoz, C Torregrosa, F-x Nuiry, K Kershaw, A Perillo-Marcone, D Grenier, R Franqueira, V Rizzoglio, A Lechner, M Calviani, Switzerland E Solfiti, F Berto, and A Alvaro. Sigraflex ® Studies for LHC CERN beam dump: Summary and perspective. *Proceedings of IPAC2021*, pages 3571–3574, 2021.
- [20] J. Maestre, C. Torregrosa, K. Kershaw, C. Bracco, T. Coiffet, M. Ferrari, R. Franqueira Ximenes, S. Gilardoni, D. Grenier, A. Lechner, V. Maire, J. M. Martin Ruiz, E. Matheson, N. Solieri, A. Perillo-Marcone, T. Polzin, V. Rizzoglio, D. Senajova, C. Sharp, M. Timmins, and M. Calviani. Design and behaviour of the Large Hadron Collider external beam dumps capable of receiving 539 MJ/dump. *Journal of Instrumentation*, 16(11), 11 2021.
- [21] J Maestre, C Torregrosa, J M Martin, K D Kershaw Senajova, M Ferrari, M Calviani, V Rizzoglio, A Lechner EN-STI, C Bracco TE-ABT, P Serrano EN-SMM, P D Pastuszak, J Guardia EN-MME, O Aberle, A Bertarelli, G Bregliozzi, A Dallochio, M Di Castro, J-f Fuchs, M Guinchard, A Lechner, S Marsh, F-x Nuiry, A Perillo-Marcone, I Ruehl, S Sgobba, H Vincke, F Carra, K Kershaw, M Timmins, C Bertone, C Garion, W M Bartmann Calviani S Gilardoni M Lamont, P Collier, M Jimenez, B Goddard, R Folch, and R Losito. CERN LHC External Beam Dumps (TDE) Modifications and upgrades during LS2 for Run3. Technical report, 2021.
- [22] Mitsuhiro Makino Dynapower corp. Improvement of domain decomposition of LS-DYNA MPP R7 and R8. Technical report, 2015.
- [23] Maurizio Vretenar. Linac4 design report Editor: Maurizio Vretenar. 2020.
- [24] Saeed Moaveni. *Finite element analysis theory and application with ANSYS*. Pearson, 2015.
- [25] Nicola Solieri and Marco Calviani. Lessons learnt from material studies, HiRadMat-56 in-beam experiment, and the autopsy of a RUN2 operational dump for the design of the HL-LHC Dumps (TDE). pages 5–5, 2022.
- [26] E Oberg, F.D Jones, Horton H.L, and Ryffel H.H. *ANSI/ASME B4.1 - Preferred Limits and Fits for Cylindrical Parts*. 2004.
- [27] P D Pastuszak, / Mme, M Timmins, En / Mme, and F Carra. Finite Element Analysis of the welds in the LHC-TDE. Technical report, 2019.



CHAPTER BIBLIOGRAPHY

- [28] Thomas Sven Pettersson and P Lefèvre. *The Large Hadron Collider conceptual design*. 1995.
- [29] Roessle M. and Fatemi A. *Strain-controlled fatigue properties of steels and some simple approximations*, volume 22. 2000.
- [30] Francesc Salvat Pujol. An introduction to the Monte Carlo method for the simulation of radiation transport. 2022.
- [31] Bhavani V Sankar, Ashok V Kumar, and Nam H Kim. *Introduction to finite element analysis and design*. John Wiley & Sons, 2018.
- [32] Shen R. Wu and Lei Gu. *Introduction to the Explicit Finite Element Method for Nonlinear Transient Dynamics*. John Wiley & Sons, Incorporated, 2012.
- [33] E. Solfiti, M. Calviani, A. Perillo-Marcone, J. M. Heredia, C. Torregrosa, A. Alvaro, and F. Berto. Flexible graphite as beam dumping material in the TDE blocks of the large hadron collider. In *Procedia Structural Integrity*, volume 28, pages 2228–2234. Elsevier B.V., 2020.
- [34] E Solfiti, Wan Di, A Celotto, N Solieri, P A Munoz, R F Ximenes, A Perillo Marcone, F X Nuiry, A Alvaro, F Berto, and M Calviani. Graphical Abstract FIB-SEM investigation and uniaxial compression of flexible graphite. Technical report, 2022.
- [35] Nicola Solieri. Dynamic studies of multiple configurations of CERN’s Antriproton Decelerator Target core under proton beam impact. 2017.
- [36] Marilena Streit-Bianchi. *CERN science bridging cultures*. 2018.
- [37] Susanna Young, Jeff Strain, and Eric Miller. *Introduction to the ANSYS Parametric Design Language (APDL)*. Phoenix analysis and design technologies, 2013.

CHAPTER BIBLIOGRAPHY

