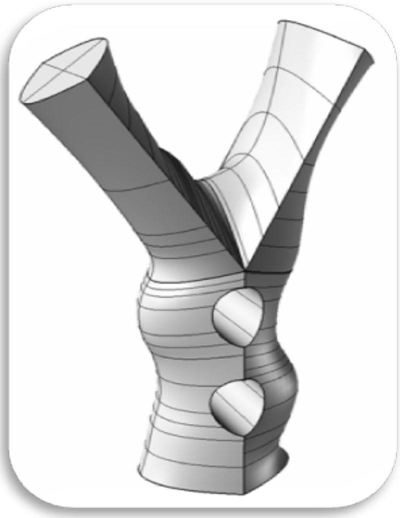
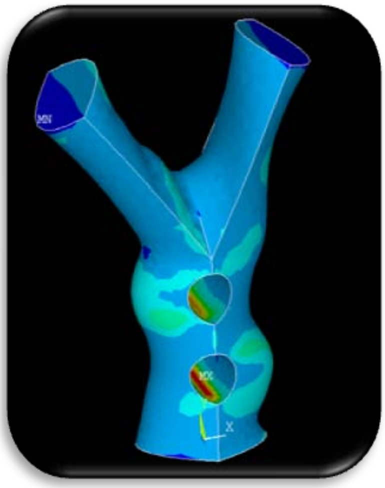
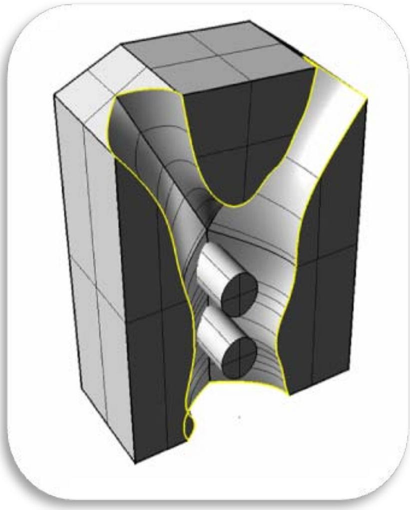




Special Nodes in Ultra High Performance Concrete

Designing unique structural nodes through computational optimisation

Master's Thesis Final Report

Hasan Han

July 2010

ARUP TUDelft

Preface

This document represents the final report of the graduation project "Special Nodes in Ultra High Performance Concrete". The graduation project was performed at the Structural Design Lab and the section Building & Structural Engineering at the Faculty of Civil Engineering and Geosciences at Delft University of Technology and in cooperation with Arup B.V. Amsterdam. The final report contains the results and products of the research project.

Amsterdam, July 2010
Hasan Han

Graduation committee:

Prof.dr.ir. Joost Walraven
ir. Jeroen Coenders
Dr. ir. Steffen Grünewald
Dr. ir. Pierre Hoogenboom

Personal & Committee Information

Personal Information:

Name:	Hasan Han
Student Number:	1142933
Date of Birth:	January 31, 1982
Home Address:	Scholtenstraat 62, 2313 GM Leiden
Telephone:	+31 6 27909219
Email:	han_hasan@hotmail.com
Work locations:	Delft University of Technology Faculty of Civil Engineering and Geosciences Room 1.95 ARUP bv Nederland Beta Building Naritaweg 118 1043 CA Amsterdam

Graduation Committee Information:

Prof. dr. ir. J.C.Walraven
Delft University of Technology
Faculty of Civil Engineering and Geosciences
Section of Structural and Building Engineering
(+31) (0)15 2785452
Room number: Stevin II-North 2.02
J.C.Walraven@tudelft.nl

Ir. J.L.Coenders
Arup B.V. Amsterdam
Delft University of Technology
Faculty of Civil Engineering and Geosciences
Section of Structural and Building Engineering, Structural Design Lab
(+31) (0)15 2785711
Room number: Stevin II-South 1.60
Jeroen.Coenders@Arup.com

Dr. ir. S.Grünewald
Hurks Beton Eindhoven
Delft University of Technology
Faculty of Civil Engineering and Geosciences
Section of Structural and Building Engineering
(+31) (0)15 2784580
Room number: Stevin II-North 1.04
S.Grunewald@tudelft.nl

Dr. ir. P.C.J.Hoogenboom
Delft University of Technology
Faculty of Civil Engineering and Geosciences
Section of Structural Mechanics
(+31) (0)15 278 8081
Room number: 6.48
P.C.J.Hoogenboom@tudelft.nl

Summary

In this thesis project it has been researched and established that it is technically feasible to create Special Nodes using Ultra High Performance Concrete (UHPC) as the construction material which contains fibre reinforcement but does not contain passive reinforcement. A Special Node is a structural element that connects incoming members in a frame structure, has a complex 3D-geometry and has geometrical distortions in the form of holes.

For this purpose a computational structural design tool that is called *VisionNode*, has been developed. *VisionNode* creates Special Nodes through computational optimisation which is performed by a genetic algorithm. The algorithm is incorporated in *VisionNode* and continuously exchanges data between a geometric-modeller, which creates the geometry of the Special Node, and a Finite-Element-Model, that determines the technical feasibility of that Special Node. The goal of the optimisation process is to create structural nodes with a complex 3D-geometry and geometrical distortions in the form of holes while optimising for material efficiency and satisfying the structural capacity of the material UHPC. Material efficiency is achieved by reducing the volume of the Special Node and the structural capacity is determined through a Mohr-Coulomb-Failure-Check. The result is an optimum design of a Special Node in UHPC that satisfies all structural checks and has a minimal volume.

Through the interactive graphical-user-interface of *VisionNode*, users can create a Special Node based on their design preferences and design modifications. Users can specify where material must be used and where no material must be used in the Special Node. This way the holes in the node can be created. *VisionNode* operates according to expectations. As a result, Special Nodes in UHPC can be created using the computational structural design tool.

The Special Nodes in UHPC are manufactured though fully automated 3D-milling of the concrete mould using polystyrene as the mould material. When the mould has been created, the UHPC mixture is poured in the mould and the Special Node is created.

Table of content

Chapter 1.

Introduction

In this chapter the gradation project will be introduced. Firstly, the research subject will be described. Secondly, the problem definition and research objectives are presented followed by the starting points and project scope. The chapter is concluded with an outline of the final report.

1.1. Special structural nodes

The building industry of today is an industry where increasingly unique and innovative structures are being designed and constructed. Advances in design software, material science and manufacture methods from the last decades, seem to have decreased the boundaries between the imaginable and what is actually constructible. Such developments have boosted innovation and produced unique structures in the building industry. Structures with complex 3D-geometrical properties like double-curved surfaces, varying angles between members and continuingly changing dimensions are but one example. The structural problems that such design would normally produce can be dealt with more efficiently today due to the advanced resources available to engineers, such as increased computational power, stronger materials and new manufacture methods.

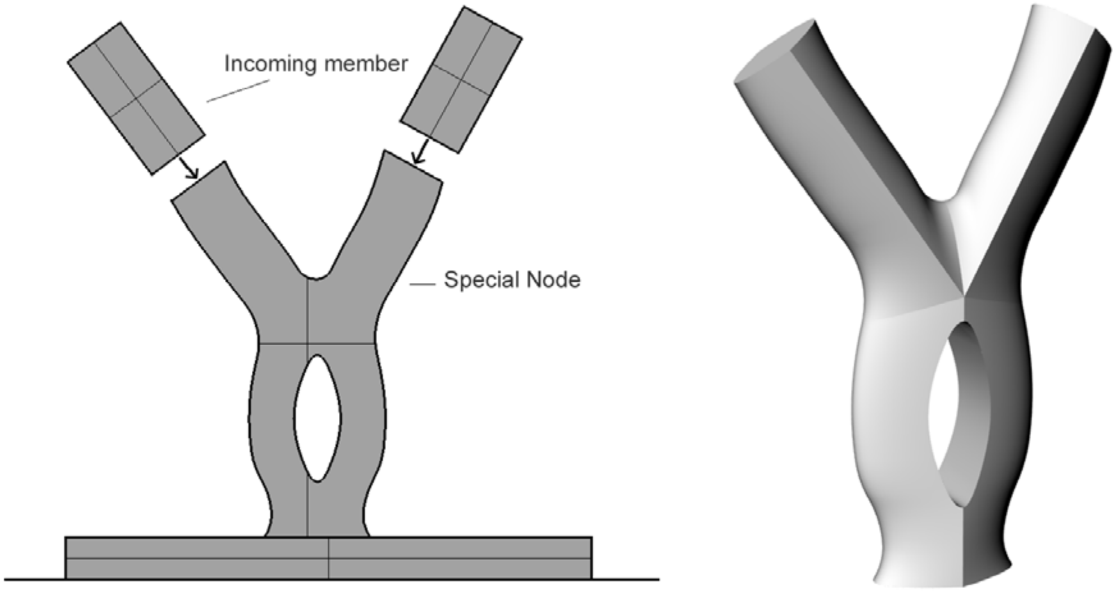


Figure 1.1. Special Node

Advancements such as these do not automatically present new possibilities and applications for design and construction. It is up to the designers and engineers to acknowledge the advanced resources and new developments and to come up with new practical applications and possibilities for the building industry. For this reason, much research is being conducted today to explore, develop and test new material, structural and computational applications.

This research project has attempted to take part in this for the case of structural nodes. Structural nodes are elements that are used in a structure to connect separate structural elements. A relatively new construction material and manufacture method has been explored in this research project to be used in a specific type of a structural node. This type, which is the subject of this research project, is called a *Special Node*. In the context of this research, a Special Node is defined as a structural element which connects incoming members in a frame structure, has a complex 3D-geometry and has geometrical distortions in the form of holes. A complex 3D-geometry can be defined as combinations of double curved surfaces, multiple sub-elements with varying angles between and continuingly varying dimensions. See Figure 1.1 to get a first impression of a Special Nodes. The element placed in the centre is the Special Node and the rectangles are the incoming structural elements.

The relatively new construction material that was mentioned earlier serves as the Special Node material and is called Ultra High Performance Concrete (UHPC).

1.2. A new construction material

UHPC is a material that has the mechanical and strength properties structural nodes require. Primarily, it has very high compressive and a relatively high tensile strengths when compared to conventional concrete. Furthermore, it is a material with excellent durability properties as well as ductility characteristics. This can be partially contributed to the presence of fibre-reinforcement in the material. Even without the addition of passive reinforcement that is normally present in concrete elements in the form of steel bars.

The superior material properties, compared to conventional concrete, make UHPC a new type of construction material. Because of this, UHPC can be applied in structural elements in new and innovative ways. For this reason, much research has been and is still being conducted to come up with new applications for the material. In the case of this research project, UHPC will be used as a construction material for Special Nodes. The application of UHPC in this way, has not been explored before and justifies this research.

The way in which UHPC is used in Special Nodes in this research, is one with a focus on freedom of design and freedom of form. To explore the technical feasibility of Special Nodes in UHPC with all sorts of shapes, angles and dimensions and without the use of passive reinforcement, is the main objective of this research. To reach this objective, the Special Nodes themselves will have to be designed and analysed first. For this purpose, it is the goal of this research project to develop a computational tool that can create Special Nodes and will serve as the means of reaching the main research objective. This tool is explained in Section 1.3.

To effectively reach the aforementioned objectives, the foremost material property of UHPC will be exploited. This property is the very high compressive strength and the relatively high tensile strength of the material. As the compressive strength of UHPC is much larger than the tensile strength, this research will focus on Special Nodes exposed to mainly compressive forces and some bending forces. As such, the material can show its true potential.

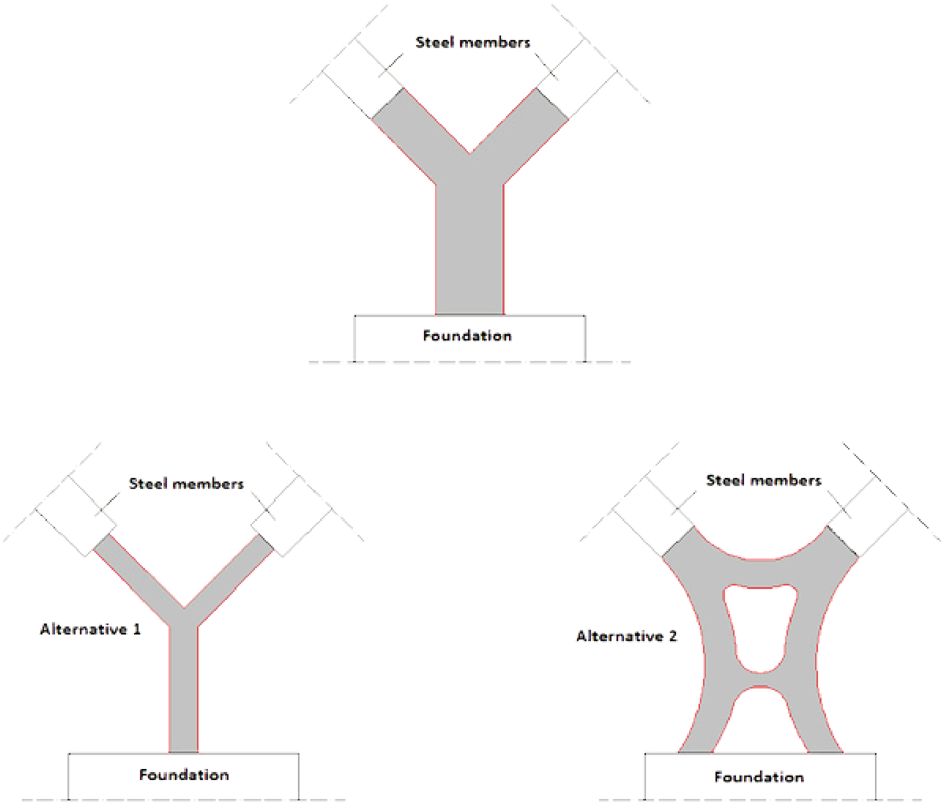


Figure 1.2. Two Design Approaches

Since UHPC has a higher structural capacity when compared to conventional concrete, two distinct design approaches for a structural node are possible. Figure 1.2 illustrates the two design approaches. The node displayed at the top of Figure 1.2 is a node made of conventional concrete exposed to a constant compressive force. The two nodes displayed underneath it, are in UHPC. The first design approach, bottom-left, is to uphold conventional methods of design and reduce the material in the structural node and simply make it as slender as possible. Because UHPC has a larger compressive capacity compared to conventional concrete, less material is required and this slendering is possible. However, aside from the obvious material reduction and slender node solution, there is also a different design approach. Making use of the higher available capacity in UHPC, the designer can chose to innovate instead of going for the most obvious solution. By applying freedom of form a certain sculptural quality can be given to the node and different designs can be created. This way an unconventional shape can be created for the node. Such a structural node is displayed at the bottom-right and can be called a Special Node. No material-reduction is achieved, but the node itself has obtained a unique and distinct new design. When innovative designs have a higher priority over cost-effectiveness, the second approach can be chosen.

Special Nodes as such, would not be suitable for conventional structures. The reason for this is that conventional construction generally involves an efficient construction-approach with emphasis on construction-speed and low-cost. An innovative structural node would mean too high an investment for conventional structures. For this reason, this research will focus on special structures and extreme structures in which Special Nodes could be applied.

The example of Figure 1.2 portrays the perspective of this research. To study the application of UHPC with fibre reinforcement and without passive reinforcement in Special Nodes. Nodes in which freedom of design and freedom of form have the overtone instead of ultimate material efficiency. The structural capacity of UHPC will be exploited to the maximum, effectively pushing the material to its limits. The way in which all this will be made possible, is through a computational structural design tool.

1.3. Computational structural design tool

The structural design process is an iterative procedure. Before a final design is decided upon, numerous preliminary designs are made and constantly altered until a final design is obtained. Because of this, the design process for conventional structural elements is time consuming. As the aim of this research project is the design of innovative structural elements in the form of Special Nodes in UHPC, the time required to produce a final design would be even more time consuming. For this reason, the processing power of computers is applied in this research, to reduce the design time.

Computational tools in engineering have been available for a number of decades. Most of these tools, however, are used for structural analysis of predetermined designs. They do not support users during the conceptual design stage and are mobilised in later design stages. As such, computational tools in this form are not sufficient for this research because the main focus is the *design* of structural elements. For this reason, a computational structural design tool is required for the very specific application of designing Special Nodes in UHPC. Since such a computational tool does not exist, one has been developed in this research project.

This computational structural design tool is called *VisionNode*. It is a Windows-application that enables users to create Special Nodes using UHPC as the construction material. First, the user creates a preliminary design with specific design properties though *VisionNode*'s interactive user-interface. Subsequently, *VisionNode* generates the final design of the structural node based on this preliminary design, through computational optimisation. The final design produced by *VisonNode,* is called a Special Node.

The heart of the computational optimisation in *VisionNode*, is a mathematical optimisation algorithm which is controlled by the tool. The algorithm used in *VisionNode* is called a *genetic algorithm* and is characterised by its global-optimisation capabilities. The optimisation process itself is essentially a continuous cycle of iterations where a certain objective-function is changed and evaluated after every iteration. The goal is to minimise the objective function. For *VisionNode,* the objective function is defined as the volume of the Special Node. In essence, the algorithm is trying to minimise the volume of the node by generating a new geometry of the Special Node during every iteration and reviewing the volume. To do this, a number of design-variables are supplied to the optimisation algorithm, which the algorithm is allowed to change every iteration until a certain optimum has been reached.

The design-variables are coordinates on the XY-plane from which the geometry of the Special Node is created. To generate a new geometry for every iteration-step, *VisionNode* uses a geometric-modeller which is a part of the program. This geometric-modeller calculates the location of the design-variables and uploads the information to the optimisation algorithm. For every geometry, the stresses in the node are calculated with a finite-element-analysis and also uploaded to the optimisation algorithm. The goal of the optimisation process, is to create the most efficient design of a Special Node by minimising the volume, and consequently the amount of material, while at the same time satisfying certain predetermined structural and geometrical restrictions and boundary conditions. The structural boundary condition consists of a maximum allowable stress in the node which are predetermined by the user. The same is also true for the geometrical boundary conditions. At the end of the optimisation process, *VisionNode* has created a Special Node that satisfies all boundary conditions and has a minimum volume. The genetic algorithm and the optimisation process are explained in more detail in Chapter 5.

Since *VisionNode* is a tool that creates Special Nodes based on a user created preliminary design, the user must have the possibility to express certain design preferences and make design modifications. This is done through the interactive graphical-user-interface (GUI) of *VisionNode*. Through the GUI, users have the opportunity to specify geometrical and structural parameters that characterise the Special Node.

These input parameters are:

- The coordinates and the dimensional properties of the incoming structural members that are connected to the Special Node
- The forces transferred from the incoming members to the Special Node
- A desired UHPC mixture with its mechanical properties and structural performance

- Specification where **no** material must be used in the Special Node, in order to create the geometrical distortions in the form of holes
- Specification where material **must** be used in the Special Node

The last two input parameters realise the design modifications. Freedom of design and form are expressed through the design modification. By explicitly specifying where material should be used and where it shouldn't be used by the geometric-modeller, the user has the means to influence the design of the node and create a Special Node with the desired characteristics.

1.4. Optimisation from a different perspective

In the previous section it has been stated that the computational structural design tool uses a computational optimisation process to find the most efficient solution. At the same time is was stated that the node that is to be optimised, contains geometrical distortions in the form of holes. These two statements seem to contradict each other since optimisation is used to obtain efficiency but at the same time the optimisation has to deal with illogical inefficiencies such as the holes. If one is looking for the most efficient solutions, it is illogical to disturb the efficiency by placing holes in a structural element. This observation would be correct for a standard situation where the goal is to find the most efficient solution to a structural problem. In this research project, the goal of the optimisation is also to find the most efficient solution to a structural problem. However, structural problem itself, the Special Node, is intentionally inefficient to begin with.

The purpose of optimisation as a concept, is viewed from a different perspective in this research. It is believed that the power of computational optimisation is that, when properly controlled and directed, it can be used to find the most efficient solution for any given structural problem. It does not matter how illogical or inefficient this structural problem initially may seem to be. This concept has been illustrated in Figure 1.2. In other words, don't look for the most obvious optimum solution, but try to find the optimum solution for a seemingly illogical structural problem. The purpose of this is to present new and innovative design possibilities. Special Nodes can also considered structurally inefficient to begin with, since they contain holes and have a complex 3D-geometry. Nevertheless, computational optimisation can still be used to create them and that is the purpose of optimisation in this research

What may have been illogical and inefficient once, does not mean that is to stay that way forever. As was explained in Section 1.1, advances in material science, computational power and manufacture methods have converted what was previously inefficient, to not so inefficient.

By combining UHPC with new manufacture methods and computational optimisation, the goal of this research project is to combine these aspects in order to make innovative and unique designs of structural nodes possible. This perspective on optimisation is the core of this research project. Through *VisionNode*, the user can direct the optimisation algorithm and force it not to generate the most simple and straightforward design of a structural node. In the hand of capable designers, *VisionNode* can be utilized to create innovative and unique Special Nodes.

To summarise, the main characteristics of *VisionNode* are formulated in the following:

- Provide users the means realise freedom of design and freedom of form for the design of a structural node
- Allow geometrical distortions in structural node in the form of holes. The holes are not limited to a single shape and can have any desired shape
- Apply computational optimisation to create Special Nodes in UHPC that have a minimum volume and structurally satisfy all checks
- Reduce design time by allowing the optimisation process to do all the hard work and still produce a technically feasible Special Node in UHPC

1.1. Problem definition

Based on the previous section the problem definition of this graduation project can be formulated as:

> *" New application possibilities for Ultra High Performance Concrete, as a construction material for structural elements, that contains fibre reinforcement but does not contain passive reinforcement, have been extensively explored in research. The application of Ultra High Performance Concrete in Special Nodes, as defined in this research and only exposed to large compressive forces, has not yet been explored. "*

A Special Node is defined as:

> *" A structural element that connects incoming members in a frame structure, has a complex 3D-geometry and has geometrical distortions in the form of holes "*

1.2. Research objectives

The primary objective of this graduation project is formulated as:

> *" To determine whether it is technically feasible to create Special Nodes, as defined in this research, in Ultra High Performance Concrete with fibre reinforcement and without passive reinforcement "*

In order to reach the main research objective a second objective is defined

> *" To develop a computational structural design tool which applies computational optimisation to create Special Nodes, while optimising for the efficient use of material "*

For the user of the tool to be able to design a Special Node, the tool must give the user a means to direct the computational design process. Therefore, a third objective can be formulated as:

> *" The user must be accommodated by the computational structural design tool to actually influence the design of the Special Node. This is accomplished by allowing the user to indicate where material <u>must be used</u> and where material <u>must not be used</u> "*

1.3. Starting points and project scope

In this section the boundary conditions and project scope are defined as:

- The research focuses on the construction material UHPC for the design of a Special Node

- This research focuses on Special Nodes exposed to compression forces primarily and additional bending- and shear forces. Special nodes are not exposed to primary external tensile forces

- This research focuses on Special Nodes in UHPC with fibre reinforcement and without passive reinforcement

- This research focuses on special– and extreme structures for application of the Special Nodes in UHPC

- This research focuses on a single structure which is used as an example structure for which a Special Node in UHPC has been created. The structure is the Yas-Hotel building in Abu Dhabi

- The Special Node in UHPC are connected to incoming concrete or steel elements. The type of connection and the proper calculations is included in the research

- For the design and calculation of the Special Node in UHPC, the French design recommendations for UHPC are used

- The Eurocode is used for load and load factor determination on the Yas-Hotel building

- The UHPC mixture Ductal C170/200 is used in this research as an example UHPC mixture for the design and calculation of the Special Node in UHPC.

- A structural node for the Yas-Hotel structure is designed and calculated manually in order to serve as a basis for comparison between the optimisation produced Special Node.

- The manufacture method of 3D-milling is used for the manufacture of the moulds for the Special Nodes

- The computational optimisation algorithm is a genetic algorithm
- The open source optimisation framework DAKOTA which contains the genetic algorithm is used for optimisation
- The computational structural design tool uses a geometric-modeller to generate different designs of the Special Nodes during optimisation. CAD-software Rhinoceros created by McNeel is part of the geometric-modeller
- The computational structural design tool uses a finite-element-analysis in order to do all the necessary structural calculations
- The used finite-element-package is Ansys created by Ansys inc

1.4. Report outline

The construction material UHPC and its properties are discussed in Chapter 2.

In Chapter 3, the structure for which the Special Node has been designed, is presented. This structure is the Yas-Hotel building in Abu Dhabi. Subsequently, a manual design of a straightforward structural node is made for this structure. The structural design has been checked using the French recommendations for UHPC.

The software architecture and program structure of the developed computational structural design tool, is explained in Chapter 4.

In Chapter 5, the applied genetic algorithm is explained. How it was implemented in *VisionNode* is also described.

The background of the finite element model used in *VisionNode* is described in Chapter 6.

The results obtained through *VisionNode* are presented and discussed detail in Chapter 7, followed by a corresponding evaluation of the virtual data and real world calculations.
The manufacture method used in this research project is explained in Chapter 8. Additionally, several examples for the connection between Special Node and the surrounding structure are also presented in this chapter.

The conclusions and recommendations of this research project are presented in Chapter 9.

Chapter 2.

Ultra High Performance Concrete

In this chapter background information on the construction material Ultra High Performance Concrete (UHPC) is presented. After a brief history concerning the development of the material, the material properties are described for the applied UHPC mixture used in this research. Subsequently, an overview of the design recommendations is presented. The chapter is finalised with conclusions.

2.1. History

Concrete is a construction material that has been used for centuries. The ancient Romans used it to build numerous structures, including the famous Pantheon. After practical knowledge concerning the use of the material was forgotten and lost for more than a thousand years, structural concrete was reinvented in the 19th century at the time of the industrial revolution. Since then it has established itself as one of the leading and most used construction materials. This is still true today.

Research and development on concrete has been ongoing ever since its reinvention. The most notable results of this research include the emergence of reinforced concrete, pre-stressed concrete, fibre reinforced concrete and self-compacting concrete.

Furthermore, in the last decades significant steps have been made in the development of higher strength concrete types in the form of High Performance Concrete (HPC), Very High Performance Concrete (VHPC) and Ultra High Performance Concrete (UHPC). The primary difference of the higher strength concrete types compared to Normal Strength Concrete (NSC), is the fact that they have higher compressive strength.

Literature presents the following designations concerning the types of concrete (de Schutters & Apers, 2007):

- Normal Strength Concrete (NSC) up to B65
- High Performance Concrete (HPC) from B65 up to B105
- Very High Performance Concrete (VHPC) from B105 up to B150
- Ultra High Performance Concrete (UHPC) from B150 and higher

As C-classes are the new denotation for concrete classes in the Eurocode, the denotation in B-classes is somewhat old. Nevertheless, the list above gives a clear impression of the different concrete types and their corresponding compressive strengths.

UHPC, also known as Reactive Powder Concrete, is a promising construction material that was first development in the early 1990's by researchers at the company Bouygues S.A. (Richard & Cheyrezy 1995). As it is a fairly new construction material, it has not been used often. This is quite natural for any new construction material which still has to prove itself. Because of this, it has seen limited practical application to this day.

Nevertheless, the material has great potential and new applications for it are being explored. For this reason research into the material is being conducted all over the world as the full potential of UHPC has yet to be discovered.

2.2. Properties of UHPC

Concrete is a mixed compound that consists among other materials of cement, aggregate and water. The way in which the mix is prepared and what type of cement, aggregate and supplementary materials are used, determine the physical, mechanical and durability qualities of concrete. UHPC is a type of concrete with a very specific mixture design which gives it superior qualities when compared to normal strength concrete. The main properties of UHPC are compressive strength up to 250 MPa, a relatively high tensile strength and a remarkable increase in durability compared even with HPC. In combination with a sufficiently high amount of steel fibres it is now possible to design sustainable lightweight concrete constructions without any additional reinforcement (Schmidt and Fehling, 2005).

UHPC differs from HPC by (Resplendino & Petitjean 2003):

- its compressive strength which is systematically higher than 150 MPa
- the systematic use of steel fibres and also polymer fibres
- its high binder content and its special selection of aggregates

2.2.1. Mechanical properties

In order to understand how UHPC obtains its superior mechanical properties, the main factors that are of influence will be identified and explained in the following. The main factors are:

- Homogeneity of the concrete mixture
- Packing density and water-cement-ratio
- Microstructure of the concrete mixture
- The addition of fibre reinforcement

A. Homogeneity of the concrete mix

Conventional concrete is a non-homogeneous material which is composed of cement-paste and aggregates, each with a different strength and modulus of elasticity. The aggregate material is usually stronger than the cement-matrix. Because of the difference in the stress-strain-relationship between the sub-materials, the stress build-up is different. For a given strain, the stress in the sub-material with a higher modulus of elasticity is larger than in the sub-material with a lower modulus of elasticity. These stress variations cause a yield pattern where the stronger sub-material yields before the weaker sub-material. As a result, peak stresses occur and the overall strength of the non-homogeneous material is reduced.

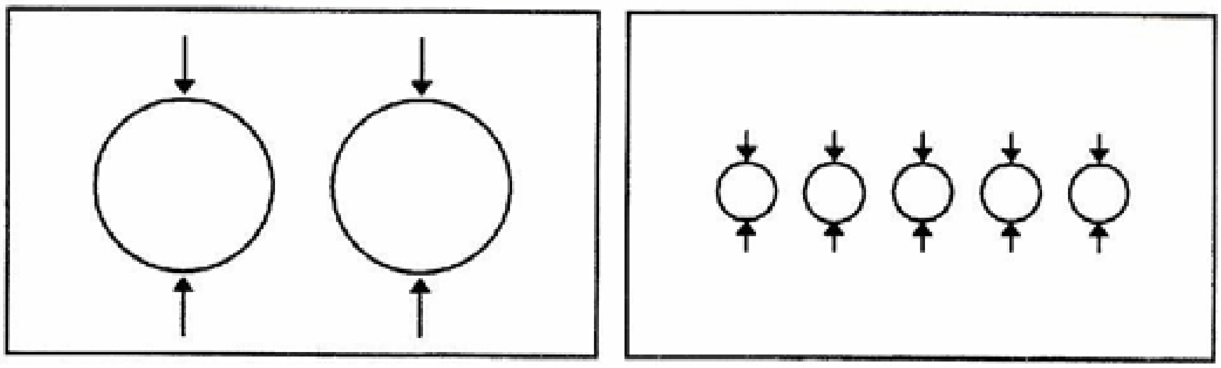


Figure 2.1. Smaller peak stresses and a more homogeneous behaviour because of smaller aggregate material (Rijkswaterstaat, 1999).

UHPC has a higher homogeneity compared to conventional concrete. This is achieved by replacing the coarse aggregate material with a much finer aggregate material. The result is a mixture with more homogeneous behaviour and much smaller stress variations and peak stresses. See Figure 2.1 and 2.2. Micro-cracks in the concrete are kept to a minimum because of this. It was suggested that the maximal aggregate size in UHPC should be less than 600 μm (Richard & Cheyrezy 1995).

In order to obtain a high strength material, the fine aggregate material has to be strong as well.

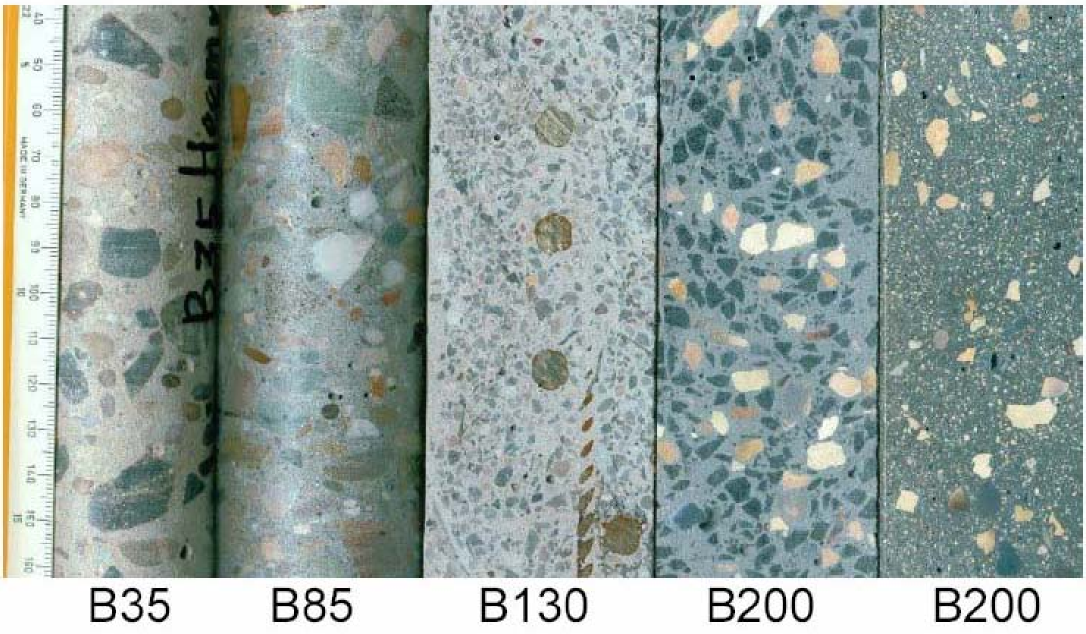


Figure 2.2. Different concrete mixtures with different size aggregate material. From left to right: B35, B85, B130/B150 (Contec), B200 (Densit) and B200 (BSI). (Kaptijn, 2002)

B. Packing density and water-cement ratio

By increasing the packing density of the mixture, the material becomes less porous and the amount of voids between the particles is reduced. These voids are normally filled with air and water and do not contribute to the strength of the material. By adding very fine material with different particle diameters to the mixture, a more dense and compact material is formed and the voids are filled with material that do contribute to the strength. As a result, the overall material becomes stronger. See Figure 2.3 and 2.4. To optimise the packing density, specific quartz powders are used with a diameter smaller than 42 μm mixed together in different amounts (Geisenhanslüke and Schmidt, 2004). Microsilica with a particle diameter of 0,1-1,0 μm is also used to obtain a more dense mixture.

The compressive strength can also be improved by using a low water-cement ratio. In conventional concrete the water-cement ratio is usually not less than 0,4 in order to obtain a good hydration and workable concrete mix. In UHPC a very low water-cement-ratio of about 0,20 to 0,25 is applied to acquire a more dense and strong material structure. By doing so the number of capillary pores is reduced due to the small amount of water in the mixture. However, a side-effect of a low water-cement-ratio is that the workability of the concrete mixture decreases. This can be compensated by applying super-plasticisers which improve the workability of a concrete mixture.

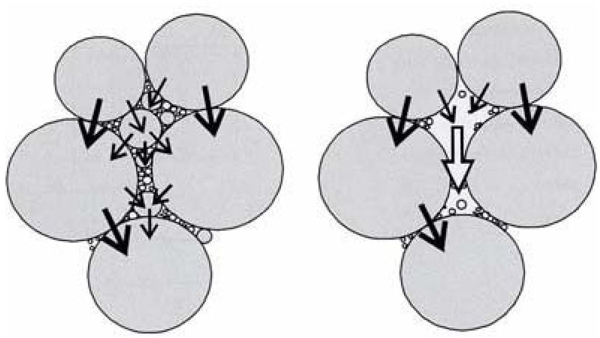


Figure 2.3. Force-interaction in UHPC left and force-interaction in NSC right. (Zimmerman, 2002)

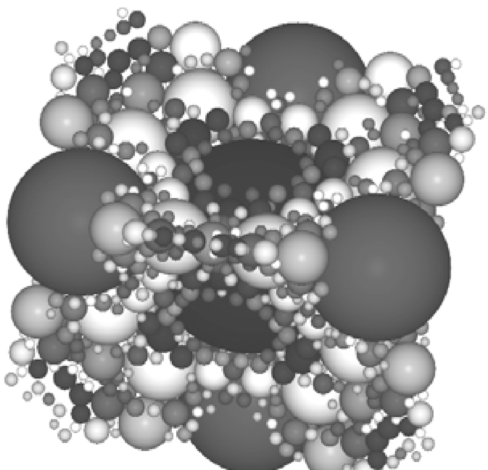


Figure 2.4. Packing effect schematical (Schmidt and Fehling, 2005).

Material	Amount (kg/m^3 (lb/yd^3))	Percent by Weight
Portland Cement	712 (1,200)	28.5
Fine Sand	1020 (1,720)	40.8
Silica Fume	231 (390)	9.3
Ground Quartz	211 (355)	8.4
Superplasticizer	30.7 (51.8)	1.2
Accelerator	30.0 (50.5)	1.2
Steel Fibers	156 (263)	6.2
Water	109 (184)	4.4

1 kg/m^3 = 1.686 lb/yd^3
RDM = relative dynamic modulus

Table 2.1. Typical composition of UHPC (Graybeal, 2006)

C. Microstructure of the concrete mixture

By post-set heat-treatment of the concrete, the microstructure of the mixture can be improved. The curing of UHPC is conducted at temperatures between 20 °C and 90 °C, 24 or 48 hours after casting. The main effect of the heat treatment is the acceleration of the hydration processes by enhancement of the pozzolanic reaction of silica fume and fine fillers (Cwirzen et al. 2008). As a result, a more dense and durable concrete mix is obtained. Heat-treated UHPC has a higher compressive and tensile capacity, the effects of shrinkage and creep are negligible. Because of the very low water-cement-ratio UHPC exhibits no post-treatment shrinkage.

D. The addition of fibre reinforcement

UHPC is a material with a high compressive strength between 150 MPa and 250 MPa. This is partially due to a high packing density and low water-cement-ratio. However, these properties not only make the material stronger, they also make it more brittle. By using steel or other adequate fibres with a high modulus of elasticity of more than 45.000 MPa, the compressive strength is kept constant while the tension, the bending tension and shear strength as well as the ductility are significantly improved (Schmidt and Fehling, 2005). The brittleness of the material is hereby reduced.

The amount of fibres used in the mixture, determines the degree of improvement of bending, tension, shear and ductility. The orientation and distribution of the fibres within the matrix are also of importance to these improvements. The most important aspect of fibre reinforcement, in the context of this research project, is the increased tensile capacity of the UHPC mixture.

Figure 2.5. Left, high strength steel fibres L/D = 6/0,15 mm; right, steel fibres L/D = 12,5/0,4 mm (www.bekaert.com).

To give an overview of the mechanical properties of UHPC compared to conventional concrete and HPC, the following is presented in Table 2.2.

	property	*range*			*unit*
		Concrete	**HPC**	**UHPC**	
		B15 to B65	C53/65 to C90/105		
Characteristic cube strength	f'_{ck}	15-65	65-105	70-800	N/mm² or MPa
Compressive strength	f'_{b}	9-39	39-60	N/A-520	N/mm²
Tensile strength	f_{b}	0,90-2,15	2,15-2,55		N/mm²
Average tensile strength	f_{bm}	1,8-4,3	4,3-5,1	6-300	N/mm²
Modulus of elasticity	E'_{b}	26.000-38.500	38.500-40.100	34.700-100.000	N/mm²
Yield strain	ε'_{bpl}	1,75	1,75-1,90		‰
Ultimate strain	ε'_{bu}	3,50	3,50-2,50		‰
Poisson's ratio	υ	0,1-0,2		0.18-0.28	-
Thermal expansion	α	9-12			$10^{-6}K^{-1}$
Thermal conductivity	k	1,8-2,5			W/mk^{-1}
Density	ρ	2500	2500	2500-4000	kg/m³
Fracture energy		130	140-150	150-40.000	J/m² or N/m

Table 2.2. Overview of mechanical properties (NEN6720; CUR Aanb. 97; Richard & Cheyrezy 1995).

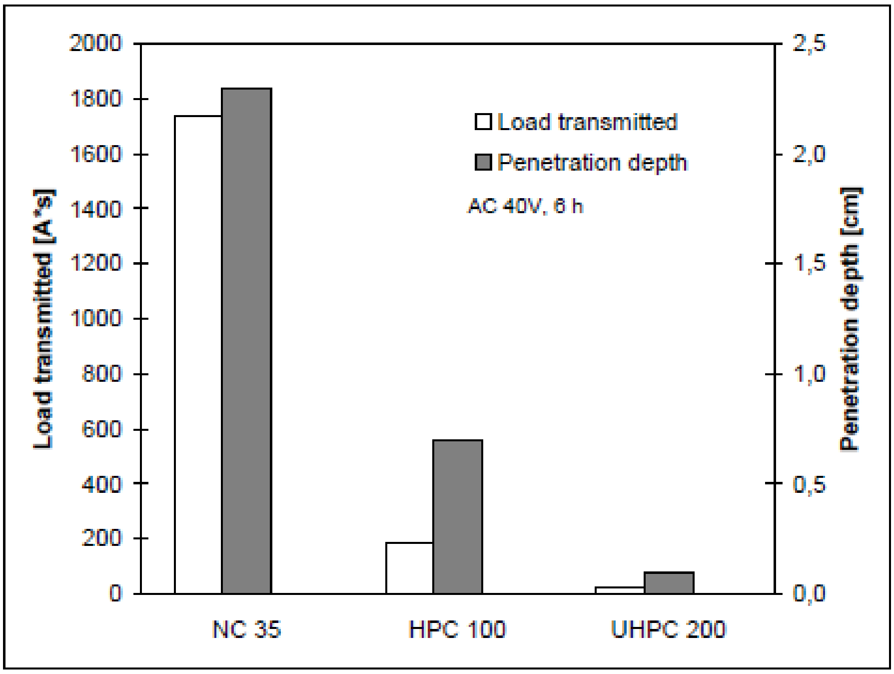


Figure 2.6. Penetration depth and transmitted loads (Teichmann and Schmidt, 2004)

2.2.2. Durability properties

UHPC has excellent durability qualities. It has a high resistance to all kinds of harmful gases and liquids, to chloride and frost and to freezing and thaw attacks. This is all related to the improved density of both the grain structure of the matrix and the much denser contact zone between matrix and aggregates, as well as the denser structure of the hydration products (Schmidt and Fehling, 2005).

The high resistance against the aforementioned types of attack is shown in Figure 2.6 by means of penetration depths between conventional concrete, HPC and UHPC. With an average penetration depth of 1,0 mm, the penetration of substances into UHPC is almost negligible. The practical benefits of this can be:

- Minimal concrete coverage
- No additional penetration prevention for the structural element
- Low maintenance of the structural element

High durability is one of the main characteristics of UHPC. In Table 2.3 an overview is presented of the durability properties of UHPC and other types of concrete.

	OC	HCP	VHPC	UHPC
Water porosity (%)	12 – 16	9 – 12	6 – 9	1,5 – 6
Oxygen permeability (m^2)	10^{-15}–10^{-18}	10^{-17}	10^{-18}	$< 10^{-19}$
Chloride–ion diffusion factor (m^2/s)	$2*10^{-11}$	$2*10^{-12}$	10^{-13}	$2*10^{-14}$
Carbonatation depth 28 days (**mm/j**)	10	2	1	<0,10
Freeze–thaw resistance (%) (**Young's Modules after 300 cycli**)	50	90	100	100
Abrasion (%) (**relative volume–loss**)	4	2,8	2	1,2
Reinforcement corrosion speed (**µm/j**)	1,20	0,25	–	<0,01
Electrical resistance (**kW*cm**)	16	96	–	1133

Table 2.3. The durability properties of different types of concrete (AFGC / SETRA (2002).

2.3. UHPC mixture Ductal

There are several commercial mixtures of UHPC available on the market today. In this research the UHPC mixture Ductal is focused upon.

Ductal is a UHPC mixture developed by LaFarge, Bouygens and Rhodia (Acker and Behloul, 2004). It is characterised by very high compressive strength, durability, flexural resistance with ductility and aesthetics. Ductal has been used in a number of applications worldwide. Among these are the Sherbrooke footbridge 1997 in Canada, presented in Figure 2.7, the Seonyu footbridge 2002 in Korea, presented in the Figure 2.8, and the Sakata Mirai footbridge 2002 in Japan, presented in Figure 2.9.

As a UHPC mixture, Ductal has very high compressive and bending strengths. The ultimate bending strength is twice its first crack stress and more than five times the ultimate stress of ordinary concrete. Such very high strength and consequent ductility allow structures without any passive reinforcement and no shear reinforcement to be designed. In Figure 2.10 the compressive strength and bending strength are displayed.

Figure 2.7. Sherbrooke footbridge in Canada (Acker and Behloul,2004).

Figure 2.8. Seonyu footbridge in Korea (Rebentrost and Wight, 2004).

Figure 2.9. Sakata Mirai footbridge in Japan (Rebentrost and Wight, 2004).

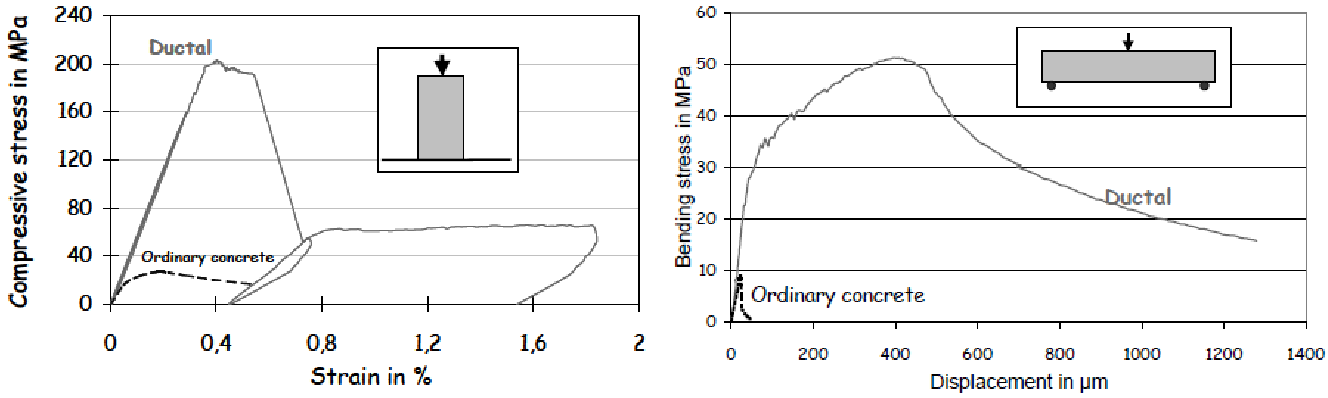


Figure 2.10. Compressive strength and bending strength of Ductal compared to ordinary concrete (Acker and Behloul, 2004).

Heat-treatment is applied to the Ductal mixtures after setting. This enables hardening at a higher rate and gives the material dimensional stability immediately after the manufacturing process. The heat treatment is performed by subjecting the mixtures to temperatures between 60 and 90°C for 48 to 72 hours. As a result, the compressive and tensile strengths are increased.

The specific Ductal mixture that will be used in this research is fibre-reinforced Ductal C170/200 which has been heat-treated. The mixture properties are presented in Table 2.4. The mechanical properties of this specific mix are presented in Table 2.5.

Element	Amount (kg/m^3)
Cement CEM I 52.5R HES	710
Quartz powder	210
Silica fume	230
Superplasticizers	13
Water	140
Steel Fibres (2.0 vol%)	160
Sand	1020

Table 2.4. Mixture Composition Ductal C170/200 (Lafarge)

Property	Value	Unit
Weight	2500	Kg/m^3
Compressive Strength	200	N/mm^2
Compressive Strength (Characteristic)	170	N/mm^2
Mean Flexural Strength	45	N/mm^2
Mean Tensile Strength	10	N/mm^2
Young's Modulus	60000	N/mm^2
Poisson Ratio	0.2	
Shrinkage Factor	550	μm/m
Creep Factor	0.3	N/mm^2
Thermal expansion coefficient	11.8	μm/m/^{0}C
Water-Cement-Ratio	0.2	
Maximum Aggregate Size	0.6	mm
Fibre Length	13-15	mm
Fibre Diameter	0.2	mm
Fibre Content	160	Kg/m^3

Table 2.5. Mechanical Properties Fibre-Reinforced Ductal C170/200 Heat-Treated (LaFarge)

2.4. Design recommendations for UHPC

At the moment, there are no official design codes available for UHPC. There are, however, two design guidelines that provide the necessary information to fully design and calculate a structural member in UHPC. These are the French recommendations for UHPC, established by AFGC/SETRA in France in 2002 (SETRA-AFGC, 2002), the German recommendations for UHPC, part of the state-of-the-art report of the DAfStB in Germany in 2003 (DAfStB UHPC, 2003), and the Japanese recommendations for UHPC (Japanese Society of Civil Engineers,2008).

As the German recommendations for UHPC are more of an addition to the French recommendations than a separate new set of design guidelines, this research will use the French recommendations for UHPC. Since the French Recommendations are used, there is no need to apply the Japanese recommendations as well. This was recommended by thesis committee members. The items from the French recommendations, relevant for this research, are described in Chapter 3.

2.5. Benefit of UHPC in Special Nodes

Now that the all main characteristics of UHPC have been discussed, the application and usefulness of it for Special Nodes is explained in this section.

The most obvious characteristic of UHPC that could be valuable for Special Nodes, are the high structural strength of the material. Since the Special Nodes are primarily exposed to high compressive forces, the most prominent material strength, the compressive strength, would be used to full potential. Furthermore, is has been stated earlier that the Special Nodes will have a complex 3D-geometry and geometrical distortions in the form of holes. Even though, no direct tensile forces are transferred from the surrounding structure to the Special Node, the presence of holes and complex 3D-geometry will cause tensile stresses to occur eventually. Since UHPC has a relatively high tensile strength, the occurring tensile stresses can be dealt with.

The relatively high tensile strength is mainly attributed to the presence of fibre reinforcement in the UHPC mixture. The fibre content also increases the ductility of the material. Ductility in a structural node is very important since slight deformations and displacement are bound to occur. For this reason, adequate ductility is required and is provided by UHPC.

Next to the material strength properties of UHPC in Special Nodes, there are also the beneficial workability properties of the mixture which are needed during the manufacture of Special Nodes. Since UHPC is a concrete material, structural elements are manufactured by casting the mixture in a mould and allow it to harden. The complex 3D-geometry and the presence of the holes would normally make the casting process difficult since the concrete has to spread to all corners and edges of the mould. The self-compacting properties and improved workability of UHPC due to the presence of super-plasticisers would make the casting process less difficult. Obtaining the desired fibre-orientation, however, may prove more difficult since the fibre-orientation is mainly influenced by the casting-direction.

2.6. Conclusion

In this chapter, the relatively new construction material UHPC has been presented and its properties have been explained.

UHPC is material with high compressive strength, a relatively high tensile strengths and high durability qualities. The superior qualities of UHPC compared to conventional concrete, are mainly the result of a high-binder content of the UHPC mixture, the addition of special aggregates to the mixture and the presence of fibre-reinforcement. The presence of fibre-reinforcement provides the option to create structural elements in UHPC without the use of passive reinforcement.

UHPC was found to be a suitable construction material for Special Nodes mainly because of its high compressive capacity and tensile strength.

Chapter 3.

Structural Node Design

In this chapter a structural node in Ultra High Performance Concrete has been designed manually. After an introduction on the design approach, information on the structure for which the node has been created is presented. Subsequently, the applied design considerations are discussed which, among other aspects, contain information on the applied building & design codes. In the third section, the created and applied computer models are presented and explained. The following section contains all the structural calculations concerning the designed node. The chapter is finalised with conclusions.

3.1. Introduction

The product of this research project is a computational structural design tool that creates Special Nodes in UHPC based on user defined preliminary designs and boundary conditions. Next to the development and application of this computational structural design tool, a structural node in UHPC for the Yas-Hotel structure is designed and calculated manually.

Before the manual design can be made, a number of preliminary operations have to be carried out. First of all, the geometric dimensions of the structure of which the structural node is a part of, have to be established and computer-models of the structure have to be made. Secondly, the force distribution from the Yas-Hotel structure transferred to the node must be obtained. When the aforementioned operations have been completed, the structural node can be designed using the Eurocode and common engineering sense. After the design, the necessary structural calculations will be performed to satisfy the structural capacity of the Special Node.

The purpose of making a manual design is to:

- Obtain the connection coordinates and dimensions of the connection incoming members from the Yas-Hotel structure
- Obtain a force distribution from the Yas-Hotel structure transferred to the structural node
- Have a basis for comparison afterwards between the manually designed structural node and the *VisionNode* produced Special Node

3.2. Yas-Hotel Abu Dhabi

In the previous section, it is mentioned that the Special Node in UHPC is part of a larger structure. This structure is the Yas-Hotel in Abu Dhabi, as was described in Chapter 1 of this report. In this research project the Yas-Hote serves as the structure for which the Special Node is designed. Some information on the Yas-Hotel is presented in the following.

Completed in October 2009 and designed by Asymptote Architects, the Yas-Hotel is located in Abu Dhabi. It is a 500 room, 85.000 square meter structure and is one of the main architectural features of the ambitious 36-billion-dollar Yas Marina development and accompanying Formula 1 raceway circuit. The main feature of the design is a 217-meter expanse of sweeping, curvilinear forms constructed of steel and 5,800 pivoting diamond-shaped glass panels. See Figures 3.1 and 3.2. This grid-shell component affords the building an architecture comprised of an atmospheric-like veil that contains two hotel towers and a link bridge constructed as sculpted steel object passing above the Formula 1 track that makes its way through the building complex. The grid-shell consists of glass plate elements and steel bearing elements. It visually connects and fuses the entire complex together while producing optical effects and spectral reflections that play against the surrounding sky, sea and desert landscape.

Figure 3.1. Yas-Hotelin Abu Dhabi (Yas-HotelPress Release, 2009).

The interesting feature of this structure for this research project, are the diagonal steel members that carry the diamond shaped grid-shell. See Figures 3.2 and 3.3. The diagonal members come together in a single point on concrete elements that transfer the forces to the foundation. The concrete element is subjected to compressive loading by the diagonal members.

Figure 3.2. Diamond shaped panels on Grid shell (Yas-HotelPress Release, 2009).

Figure 3.3. The hotel under construction (Yas-HotelPress Release, 2009).

Focus is placed upon the node section of the structure displayed in Figure 3.4. This section is comprised out of two diagonal steel members converging in a single point and on top of a supporting concrete element. The elements and forces all exist in a single plane, making this a 2D problem. For the design of a structural node, this section can be modified in such a way that the diagonal steel elements do not converge into a single point. All members would be disconnected in that case, requiring a additional structural element to connect them. Now, a structural node in UHPC can be designed for this section.

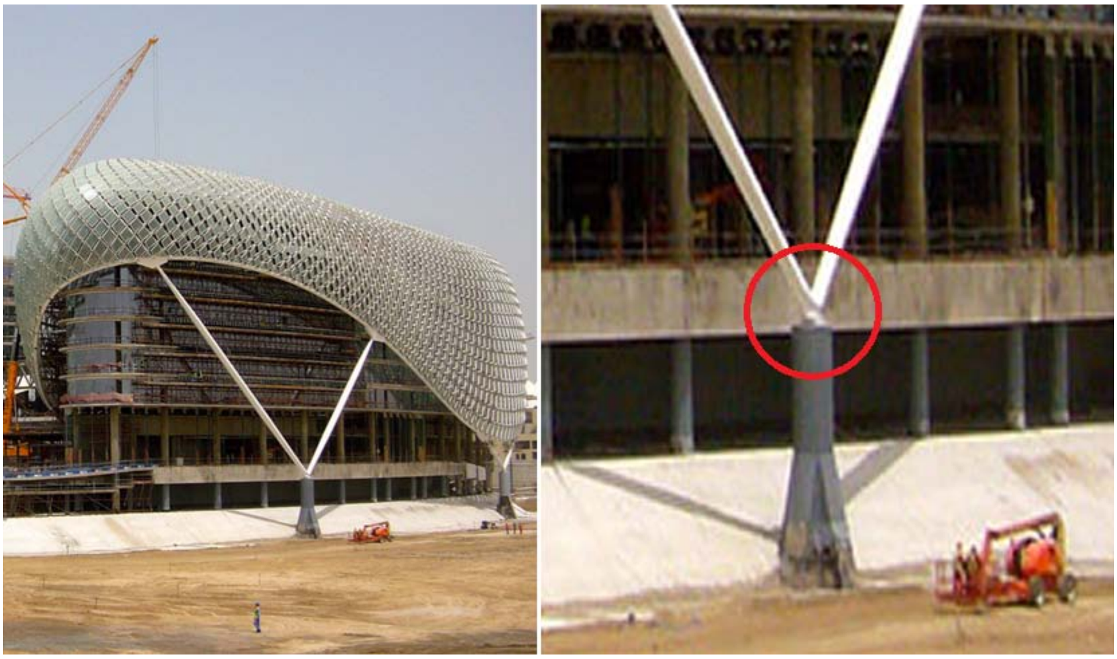


Figure 3.4. Location of the Special Nod (Yas-HotelPress Release, 2009).

As can be seen from the figures, the dimensions of the diagonal steel members and the concrete element are quite large. The exact dimensions, however, are not known as no information was available on this subject. To create a computer model of the structure, rough dimensions are satisfactory as long as the proportions between the structural elements are accurate. For this reason, a rough estimation of the dimensions has been made using the available figures. In Figure 3.5 the dimensions are estimated using the length of the person standing in the bottom-right as a basis. The dimension of each relevant element is represented with a coloured line and its corresponding coloured magnitude in the bottom-right.

The dimensions:

- Person height (light green) = 2 meters
- Structure height (light blue) = 52 meters
- Concrete column height (purple) = 9 meters
- Diagonal member A length (red) = 25 meters
- Diagonal member B length (blue) = 39 meters
- Building floor height (yellow) = 4 meters

Figure 3.5. Rough Dimension of Structure

With regard to the force interaction between the building and the grid-shell, the following assumption is made in this research project. The grid-shell surrounds the main building and transfers its vertical forces through the diagonal steel members to the foundation. The grid-shell is connected to the building with the use of horizontal braces, to obtain horizontal stability. The latter assumption is deduced from Figure 3.6.

Based on the aforementioned dimensions of the building and grid-shell members, a geometric computer model of the Yas-Hotel structure has been made. The model also includes the assumptions made in the previous paragraph. The geometric model of the Yas-Hotel structure is presented in Section 3.4

Figure 3.6. Grid-shell connection to building (Yas-HotelPress Release, 2009).

3.3. Design Considerations

In order to design a structural node in UHPC for the Yas-Hotel structure, several design considerations are stated in this section. Firstly, all the applied design codes and recommendations are described. Subsequently, the loads and load-factors on the structure are determined using the design codes followed by a short overview of the material properties of the UHPC mixture.

3.3.1. Design codes and recommendations

It must be noted that the location of the target structure of the Yas-Hotel is repositioned from Abu Dhabi to Zuid-Holland in the Netherlands. This way the Eurocode can be applied. In the continuation of this section the Yas-Hotelwill be referred to as: the structure.

In this research project the following codes and recommendations are applied

A. Eurocode:

- Eurocode 0: 1990 – Basis of Structural Design
- Eurocode 1: 1991 – Actions on Structures
    - 1-1 General actions: Densities, self-weigh and imposed loads
    - 1-3 General actions: Snow loads
    - 1-4 General actions: Wind actions
- Eurocode 3: 1993 – Design of Steel Structures
    - 1-1 General rules and rules for buildings

B. French Recommendations for Ultra High Performance Concrete

The items from French recommendations for UHPC that are relevant for this research project are the presented below. For more detailed information the reader is referred to the original document of SETRA-AFGC 2002.

- Compressive strength
- Direct tensile strength

The flexural strength and shear strength are disregarded as they are not considered in this research.

Compressive strength

The compressive strength capacity is calculated with the following equation:

$$f_{cd} = \frac{0,85 * f_{ck}}{\gamma_c}$$

f_{cd} = compressive strength
f_{ck} = characteristic compressive strength
γ_c = material factor=1,3

Direct tensile strength

The direct tensile strength capacity is calculated with the following equation:

$$f_{ctd} = \frac{0,7 * f_{ctm}}{K * \gamma_c}$$

f_{ctd} = tensile strength
f_{ctm} = mean tensile strength
γ_c = material factor=1,3
K = fibre orientation factor

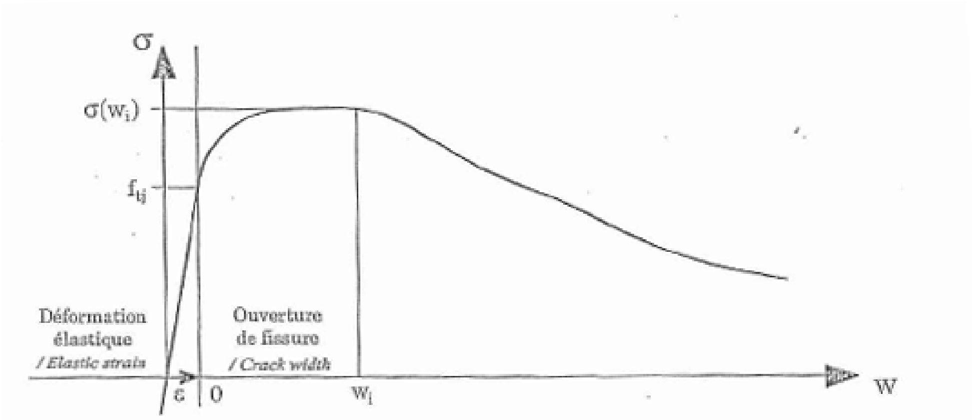


Figure 3.6. Example of tensile constitutive law of UHPC (AFGC / SETRA ,2002).

According to the French recommendations, the tensile behaviour of UHPC is characterized by:

- An elastic stage limited by the tensile strength of the cement matrix f_{tj}
- A post-cracking stage characterised by the tensile strength of the composite material after the matrix has cracked.

The effective tensile strength of UHPC is partially determined by the amount of fibres in de mix and the fibre orientation. The latter is mostly determined by the method of casting and the flow of the material during casting. The recommendations take into account these phenomena and provide an 'intrinsic' curve for tensile behaviour which does not depend on test specimen size or on the type of test used. The intrinsic curve is derived from characterisation tests and corresponding transfer- and reduction-factors which are impeded in the previous formula.

3.3.2. Loads on structure

For this research project only the force distribution on the diagonal steel members, and consequently on the Special Node, is of relevance. Based on the assumptions made in Section 3.2, only the dead loads and live loads on the grid-shell are important. The loads on the internal building are not regarded since no vertical or horizontal forces are transferred from the building to the grid-shell.

A. Dead Load - Self-weight of the glass and steel elements of the grid-shell

Applied code = Eurocode 1: 1991 1-1 – Appendix A
Load of glass element = 25 kN/m^3 (Appendix A – table A.5)
Glass element thickness = 50 mm
Load = 25*0,05= 1.25 kN/m^2
Total load including steel bearing elements = 1,25+0,25 = 1,5 kN/m^2

The glass element thickness has been determined after consulting several colleagues at Arup Amsterdam. The same applies for the load including steel bearing elements. A total load of 1,5 kN/m^2 for the grid-shell, is a useable value.

B. Dead Load- Self-weight of the diagonal steel members

The self-weight of the diagonal steel members which connect to the Special Node, is calculated with the structural analysis program GSA 8.4 (www.oasys-software.com). The magnitude of the self-weight is automatically included in the structural calculations which are presented in section 3.5 of this chapter.

C. Live Load - Roof maintenance load

Applied code	= Eurocode 1: 1991 1-1 – Article 6.3.4.
Roof surface class	= H
Load	= 1,0 kN/m^2

The roof surface class H is assigned to roofs with are only accessible for maintenance and repair, as depicted in article 6.3.4. in Eurocode 1.

D. Live Load - Wind Load

Applied code	= Eurocode 1: 1-4 – Article 4
Wind region	= II (Article 4.2 – National Annex)

The wind load determination consists of several elements which are calculated in the following order:

1) Base wind velocity: v_b [m/s]
2) Mean wind velocity at height z above the terrain: $v_m(z)$ [m/s]
3) Peak velocity pressure at height z above terrain: $q_p(z)$ [kN/m^2]
4) Wind pressure: w_e [kN/m^2]

1) Base wind velocity: v_b **[m/s]**

$$v_b = C_{dir} * C_{season} * v_{b,0}$$

$v_b =$ basic wind velocity
$C_{dir} =$ directional factor (=1,0)
$C_{season} =$ season factor (=1,0)
$v_{b,0} =$ fundamental value of basic wind velocity (=27 m/s wind-region II)
(Article 4.2 - National Annex)

Basic wind velocity : $v_b = 27$ m/s

2) Mean wind velocity: $v_m(z)$ **[m/s]**

$$v_m(z) = c_r(z) * c_o(z) * v_b$$

$c_r(z) =$ roughness factor
$c_o(z) =$ orography factor (1,0) (Article 4.3)

Roughness factor:

$c_r(z) = k_r * \ln * (\frac{z}{z_0})$ for $z_{\min} \leq z \leq z_{\max}$

$z =$ height = 52 m (building height)
$z_0 =$ roughness length = 1,0 m for terrain catergory IV (Table 4.1)
$k_r =$ terrain factor dependant on the roughness length

$k_r = 0,19 * (\frac{z_0}{0,05})^{0,07}$

$z_{\max} =$ 200 m (Article 4.3)
$z_{\min} =$ 10 m for terrain catergory IV (Table 4.1)

$k_r = 0,19 * (\frac{1,0}{0,05})^{0,07} = 0,24$

$c_r(z) = 0,24 * \ln * (\frac{52}{1,0}) = 0,95$

Mean wind:

$v_m(z) = c_r(z) * c_o(z) * v_b$

$v_m(z) = 0,95 * 1,0 * 27 = 25,6$ m/s

3) Peak velocity pressure: $q_p(z)$ **[kN/m²]**

$$q_p(z) = (1 + 7 * I_v(z)) * 1/2 * \rho * v_m{}^2(z)$$

$\rho =$ air density =1,25 kg/m^3 (Article 4.5 - National Annex)

$I_v(z) =$ turbulence intensity

$$I_v(z) = \frac{k_l}{c_0(z) * \ln(\frac{z}{z_0})}$$

$k_l = 1,0$ (Article 4.4)

$$I_v(z) = \frac{1,0}{1,0 * \ln(\frac{52}{1,0})} = 0,25$$

$$q_p(z) = (1 + 7 * 0,25) * 1/2 * 1,25 * (25,6)^2 = 1,126 \text{ kN/m}^2$$

Peak velocity pressure: $q_p(z) = 1,126$ kN/m^2

4) Wind pressure: w_e **[kN/m²]**

$$w_e = q_p(z) * c_{pe}$$

$c_{pe} =$ pressure coefficient for external windpressure

$c_{pe} =$ +0,8 for surface pressure (Article 7.2.2)

$c_{pe} =$ -0,5 for surface suction (Article 7.2.2)

$w_{e;pressure} = 1,126 * 0,8 = 0,90$ kN/m^2 for pressure surface

$w_{e;suction} = 1,126 * 0,5 = 0,56$ kN/m^2 for suction surface

Wind pressure: $w_{e;pressure} = 0,90$ kN/m^2

Wind suction: $w_{e;suction} = 0,56$ kN/m^2

E. Live Load – Snow Load

Applied code = Eurocode 1: 1991 1-3

Snow load *s*:

$$s = \mu_i * C_e * C_t * s_k$$

μ_i = snowload formcoefficient

s_k = charcteristic value of snowload on ground (kN/m2)

C_e = exposure coefficient

C_t = heat coefficient

C_e = 1,0 (normal circumstances)

C_t = 1,0

μ_i = 0,8 (lessenaarroof met $\alpha = 0^\circ$)

s_k = 0,7 kN/m2 (applied to all locations in the Netherlands - 1991 1-3 Annex)

Snow load s: $s = 0,8 * 1 * 1 * 0,7 = 0,56$ kN/m^2

3.3.3. Load factors and load combinations

The load factors, formulated in the Eurocode, that apply to the current structure are presented in Table 3.1.

Load type	Permanent	Variable
ULS (favourable/unfavourable)	1,35/0,9	1,5
ULS special	1,00	1,00
SLS (favourable/unfavourable)	1,00/1,00	1,00
SLS special	1,00	1,00

Table 3.1. Load factors

The load combinations are expressed in so called momentane factors ψ_0 and are presented in Table 3.2. The factors are all zero. This means that only one at a time can be prominent.

Load type	Momentane factor ψ_0
Roof maintenance	0
Snow load	0
Wind load	0

Table 3.2. Load combinations

3.3.4. Material properties

The material properties of the fibre-reinforced UHPC mixture Ductal C170/200 which has been heat-treated, are presented in Section 2.3 of Chapter 2. For convenience, the mixture properties are also presented in this sub-section.

Element	Amount (kg/m^3)
Cement CEM I 52.5R HES	710
Quartz powder	210
Silicafume Slurry	230
Superplasticizers	13
Water	140
Steel Fibres (2.0 vol%)	160
Sand	1020

Table 3.3. Mixture Composition Ductal C170/200 (Lafarge)

Property	Value	Unit
Weight	2500	Kg/m^3
Compressive Strength	200	N/mm^2
Compressive Strength (Characteristic)	170	N/mm^2
Mean Flexural Strength	45	N/mm^2
Mean Tensile Strength	10	N/mm^2
Young's Modulus	60000	N/mm^2
Poisson Ratio	0.2	
Shrinkage Factor	550	μm/m
Creep Factor	0.3	N/mm^2
Thermal expansion coefficient	11.8	μm/m/^{0}C
Water**-Cement-Ratio**	0.2	
Maximum Aggregate Size	0.6	mm
Fibre Length	13-15	mm
Fibre Diameter	0.2	mm
Fibre Content	160	Kg/m^3

Table 3.4. Mechanical Properties Fibre-Reinforced Ductal C170/200 Heat-Treated (Lafarge)

3.4. Computer Models

3.4.1. Geometrical Model

In this sub-section the geometric model of the structure will be presented. The exact dimensions of the Yas-Hotel structure are not known as no information was available on this subject. To create a computer model of the structure, rough dimensions are satisfactory as long as the proportions between the structural elements are accurate. For this reason, a rough estimation of the dimensions is made using the available figures.

The definition of a model is a simple representation of a real object while including only the necessary attributes. This is also true for the geometric model of the Yas-Hotel structure. The model is a simplification of the real building and for this reason does not have the exact same shape like the building in the figures of Section 3.2. The geometric models are presented in Figure 3.8 through 3.10.

The model of the structure has been made with 3D CAD-software Rhinoceros. The purpose of the geometric model is to serve as a basis for the structural analysis model. For this reason, only the structure dimensions as presented in Section 3.2 and the main bearing elements of the structure have been modelled. The shape of the model is more rectangular than the real shape of the structure.

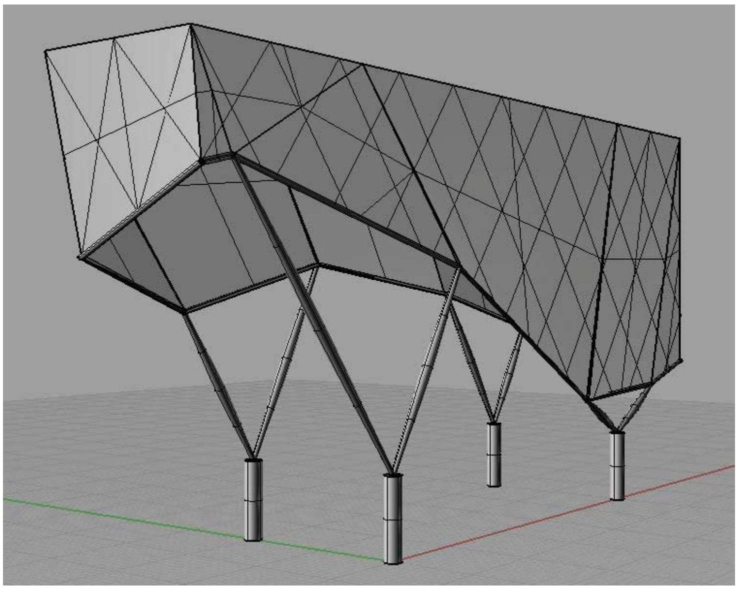


Figure 3.8. Solid Geometric Model of structure

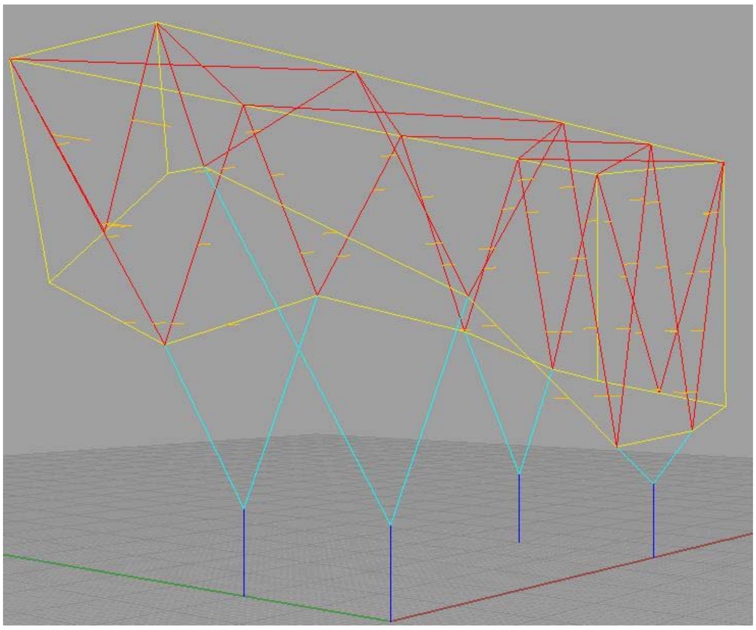


Figure 3.9. Wireframe Geometric Model of structure

In Figure 3.8 a solid model of the structure can be seen. In Figure 3.9 the wireframe model of the structure is presented and the main bearing elements are can be seen. In Figure 3.10 the wireframe model shows the composition of the main bearing elements in several steps. The concrete columns and diagonal steel members are presented in the top-left, the truss structure

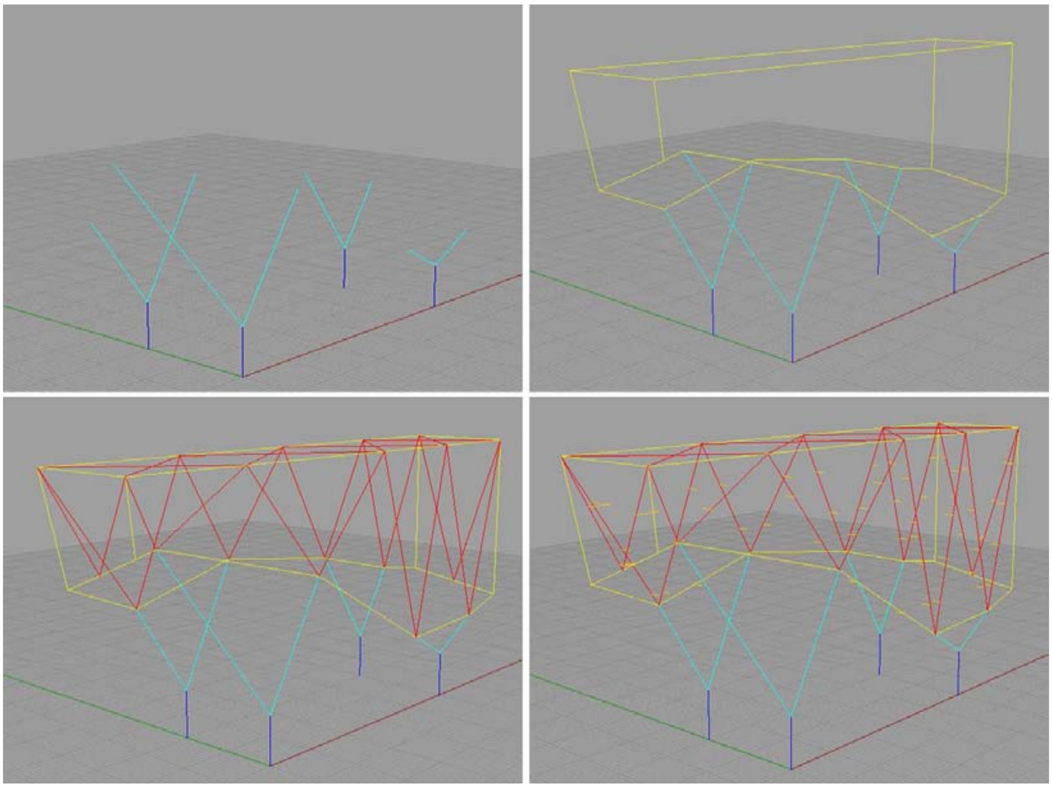


Figure 3.10. Wireframe Geometric Model composition

that carries the grid-shell is added in the bottom-left and the braces that connect the grid-shell to the main building are added in the bottom-right.

The geometry and the amount of the bearing elements has been deduced from the available figures of the structure and common engineering sense. The geometric model is converted into an structural analysis model in the next sub-section.

3.4.2. Structural Analysis Model

The structural analysis model of the structure has been made using the structural analysis program GSA 8.4 (www.oasys-software.com). GSA allows the import of DXF.file models and converts the geometric models to a structural model. The model is presented in Figure 3.11. To create an accurate structural model and prior to running the structural analysis, additional operations were performed and include the following.

A. Assign material properties to elements
B. Assign cross-section properties to elements
C. Assign restraints to supports
D. Assign loads to elements

These content of the above operations is described on the next pages.

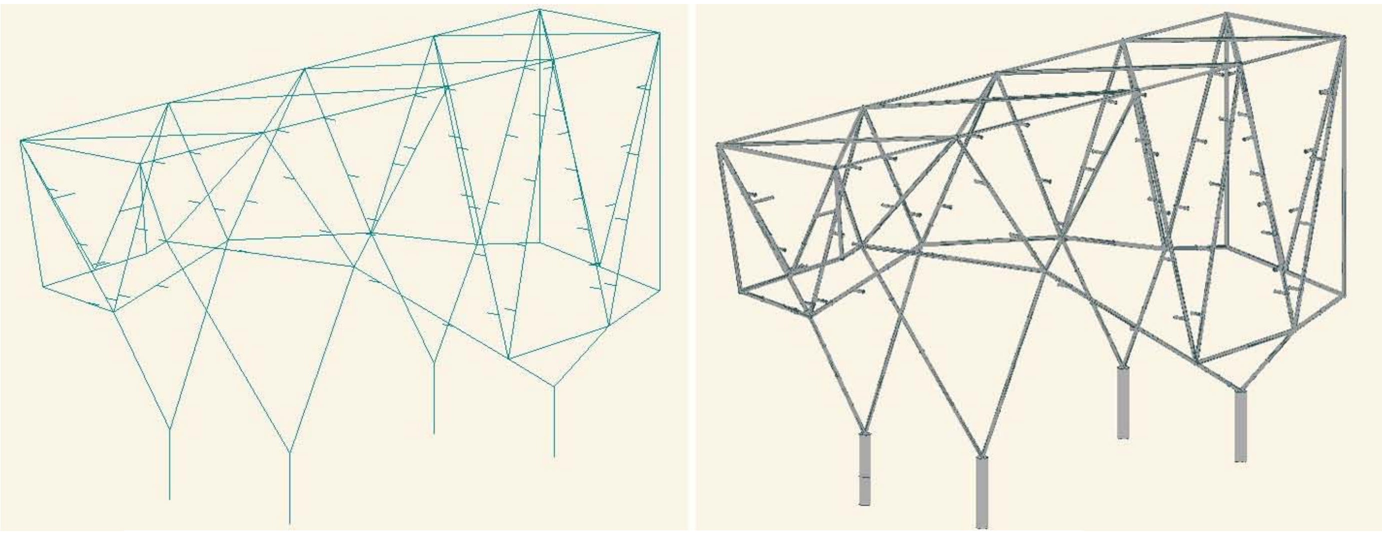


Figure 3.11. Structural model of structure. Left: wireframe representation of elements. Right: representation of the section elements

A. Assign material properties to elements

- The steel class of all the steel elements in the model is S235
- The concrete class of the columns is C45/55

B. Assign cross-section properties to elements

The cross-section properties of all the elements are based on assumptions as no data was available on these properties. The assumptions are based on the available figures of the structure and on common engineering sense. The following cross-section properties are assigned to the elements.

- Diagonal steel members
  - Hollow steel section
  - Diameter at middle = 700 mm
  - Diameter at both ends = 400 mm
  - Thickness = 10 mm
- Truss steel members
  - Hollow steel section
  - Diameter = 600 mm
  - Thickness = 10 mm
- Brace steel members
  - Hollow steel section
  - Diameter = 400 mm
  - Thickness = 10 mm
- Concrete column
  - Diameter = 1500 mm

The cross-section dimensions of the diagonal steel members are of particular importance because these are the elements that connect to the structural node in UHPC. In order to determine if the assumed cross-section dimensions of the diagonal steel members are substantiated, a structural buckling check has been made and can be found in Appendix A. The outcome of the buckling check proves that the assumed dimensions are correct.

C. Assign restraints to supports

- The concrete columns are fixed to the foundation and are restrained in all degrees of freedom
- The braces are connected to the truss elements and to the main building. The point of the connection to the main building is restrained in horizontal (X and Y) degrees of freedom

D. Assign loads to elements

The loads on the structure have been derived in Section 3.3.2 and have been applied to the structural model.

The dead load of the grid-shell elements is an area load. It has been applied to the structural model as a point load after multiplying the affected area and the magnitude of the dead load. See Figure 3.12. The live loads have been applied to the model in the same way.

The dead load of the grid-elements on roof and facade:

Roof corner	$F_{roof;1} = 1,5 \text{ kN/m}^2 * 125 \text{ m}^2 = 187 \text{ kN}$
Roof middle	$F_{roof;2} = 1,5 \text{ kN/m}^2 * 250 \text{ m}^2 = 375 \text{ kN}$
West-Facade	$F_{facade;west} = 130 \text{ kN}$
East-Facade	$F_{facade;east} = 125 \text{ kN}$
North-Facade	$F_{facade;north} = 40 \text{ kN}$
South-Facade	$F_{facade;south} = 62 \text{ kN}$

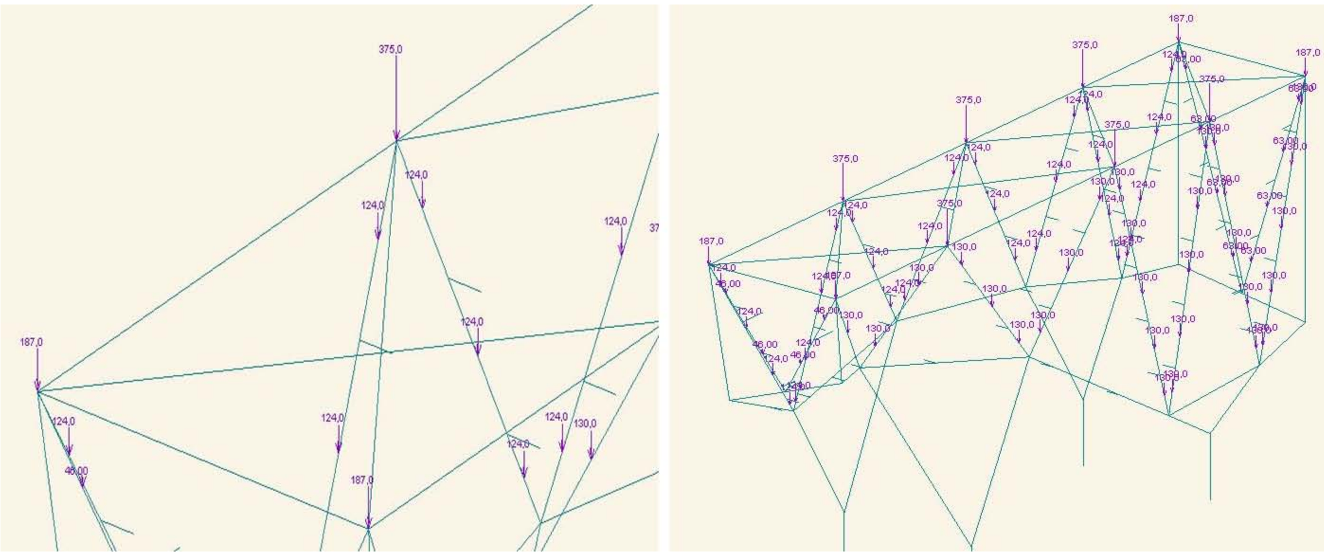


Figure 3.12. Dead load of grid-shell elements. Unit: kN

The live load of maintenance on the roof:

Roof corner $F_{maintenance;1} = 1,0\ \text{kN/m}^2 * 125\ \text{m}^2 = 125\ \text{kN}$

Roof middle $F_{maintenance;2} = 1,0\ \text{kN/m}^2 * 250\ \text{m}^2 = 250\ \text{kN}$

The live load of snow on the roof:

Roof corner $F_{snow;1} = 0,56\ \text{kN/m}^2 * 125\ \text{m}^2 = 70\ \text{kN}$

Roof middle $F_{snow;2} = 0,56\ \text{kN/m}^2 * 250\ \text{m}^2 = 140\ \text{kN}$

The live load of the wind pressure on west facade and wind suction on east facade:

Wind pressure west facade $F_{wind;pressure} = 78\ \text{kN}$

Wind suction east facade $F_{wind;suction} = 47\ \text{kN}$

The values of the above point loads are representative values. For calculations in the serviceability limit state (SLS) and ultimate limit state (ULS), the load-factors from Table 3.1 are applied.

After completing the structural model, the force distribution has been calculated it can be found in Section 3.5.

Force distribution in structural node

The force distribution on the structural node is presented through graphical and numerical output obtained from the structural GSA model.

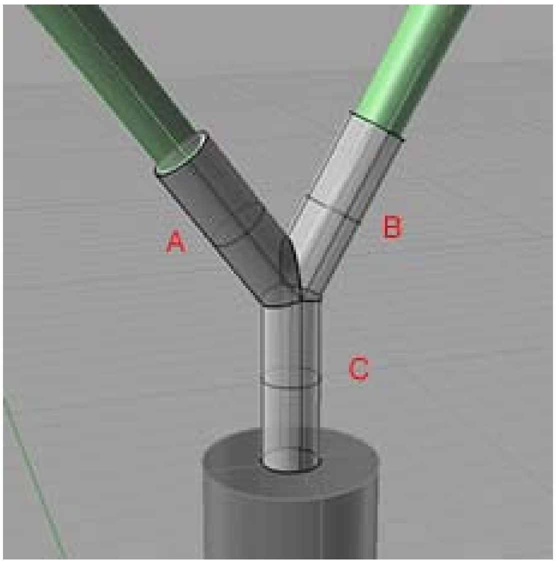


Figure 3.13. Structural node arms

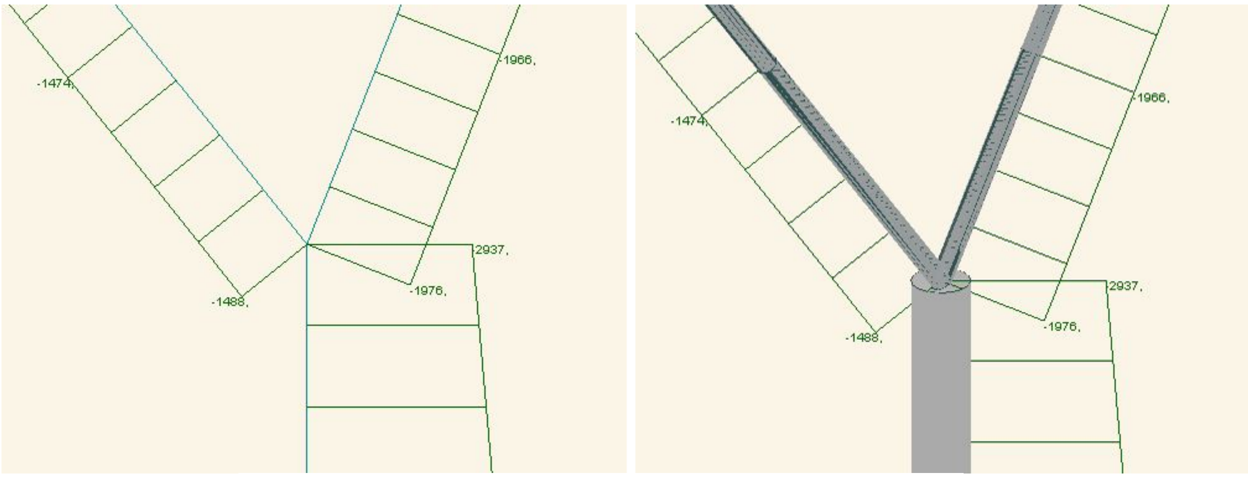


Figure 3.14. Axial stress distribution in node

A. Axial Stress

See Figure 3.14.

Axial force in arm A: N_x= –1488 kN (compression)

Axial force in arm B: N_x= –1976 kN (compression)

Axial force in arm C: N_x= –2937 kN (compression)

Axial stress in arm A: $$\sigma_x = \frac{F}{A} = \frac{-1488*10^3}{\pi*(\frac{500}{2})^2} = -7,57 \text{ N/mm}^2$$

Axial stress in arm B: $$\sigma_x = \frac{F}{A} = \frac{-1976*10^3}{\pi*(\frac{500}{2})^2} = -10,06 \text{ N/mm}^2$$

Axial stress in arm C: $$\sigma_x = \frac{F}{A} = \frac{-2937*10^3}{\pi*(\frac{500}{2})^2} = -14,95 \text{ N/mm}^2$$

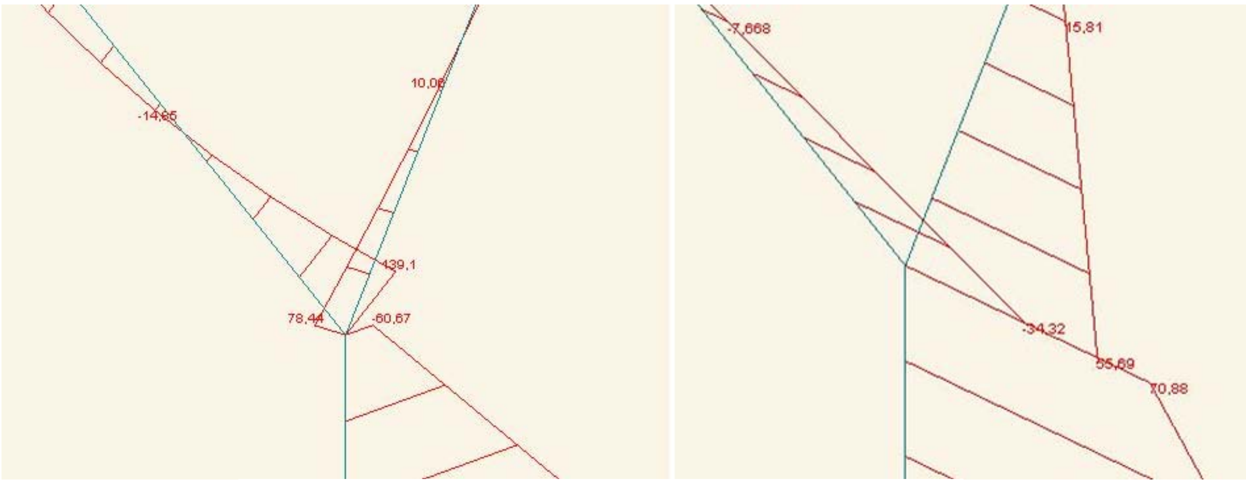


Figure 3.15. Bending stress distribution in node

B. Bending Stress

See Figure 3.15.

Bending moment in arm A: M_y= 139,1 kNm M_z= 34,32 kNm

Bending moment in arm B: M_y= 78,44 kNm M_z= 55,09 kNm

Bending moment in arm C: M_y= 60,67 kNm M_z= 70,88 kNm

Sectional modulus for circular cross-section: $W_{el} = \frac{\pi * d^3}{32} = 122,7*10^5$

Bending stress in arm A: $\sigma_y = \frac{M}{W_{el}} = \frac{139,1*10^6}{122,7*10^5} = 11,32 \text{ N/mm}^2$

Bending stress in arm B: $\sigma_y = \frac{M}{W_{el}} = \frac{78,44*10^6}{122,7*10^5} = 6,39 \text{ N/mm}^2$

Bending stress in arm C: $\sigma_y = \frac{M}{W_{el}} = \frac{60,67*10^6}{122,7*10^5} = 4,94 \text{ N/mm}^2$

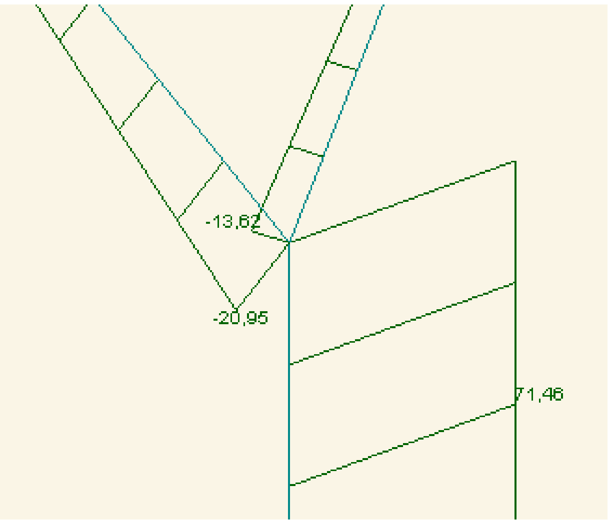


Figure 3.16. Shear stress distribution in node

C. Shear stress

See Figure 3.16.

Shear force in arm A: F_z= 20,95 kN

Shear force in arm B: F_z= 13,62 kN

Shear force in arm C: F_z= 71,46 kN

Shear stress in arm A: $$\tau_d = \frac{F_z}{A} = \frac{20,95*10^3}{\pi*(\frac{0,9*500}{2})^2} = 0,132 \text{ N/mm}^2$$

Shear stress in arm B: $$\tau_d = \frac{F_z}{A} = \frac{13,62*10^3}{\pi*(\frac{0,9*500}{2})^2} = 0,086 \text{ N/mm}^2$$

Shear stress in arm C: $$\tau_d = \frac{F_z}{A} = \frac{71,46*10^3}{\pi*(\frac{0,9*500}{2})^2} = 0,45 \text{ N/mm}^2$$

3.4.3. Structural Node Design Model

The dimensions of the adjacent diagonal steel members and the magnitude of the forces on the structure, have been taken into consideration when designing the structural node in UHPC. The manual design of the node is straightforward and practical. In Figure 3.17 the design of the structural node in UHPC is presented.

The structural node has a circular cross-section and has the following dimensions:

- Diameter cross-section = 500 mm
- Node arm length (arm a,b,c) = 1500 mm
- Total node height = 2810 mm
- Total node width = 1610 mm

The diameter of the cross-section of the node is chosen as such, because the diameter of the adjacent diagonal steel member is 400 mm at the end. This way, there is still room to facilitate a connection between the steel member and the node when applying a bolted connection.

The node has been designed by setting a circle with a radius of 1,5 m in the centre of the original connection between the diagonal steel members and the concrete column as presented in Figure 3.18. Within the boundaries of the circle, the structural node is placed. See Figure 3.19 for this. The node is not symmetrical as the angles between the diagonal steel members on the left and right are different.

The structural node in UHPC, as it is designed and presented in this section, is structurally checked in the next section of this chapter.

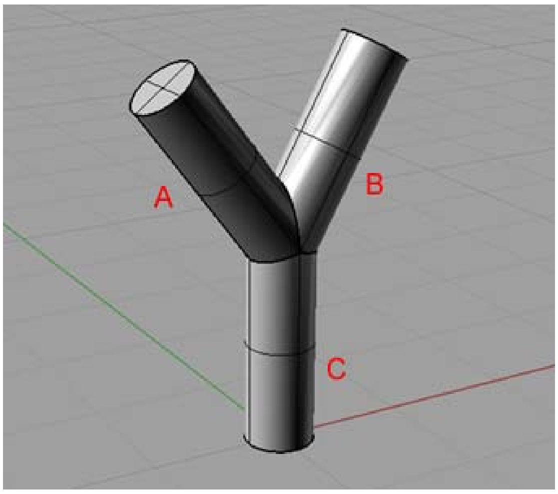


Figure 3.17. Structural node in UHPC

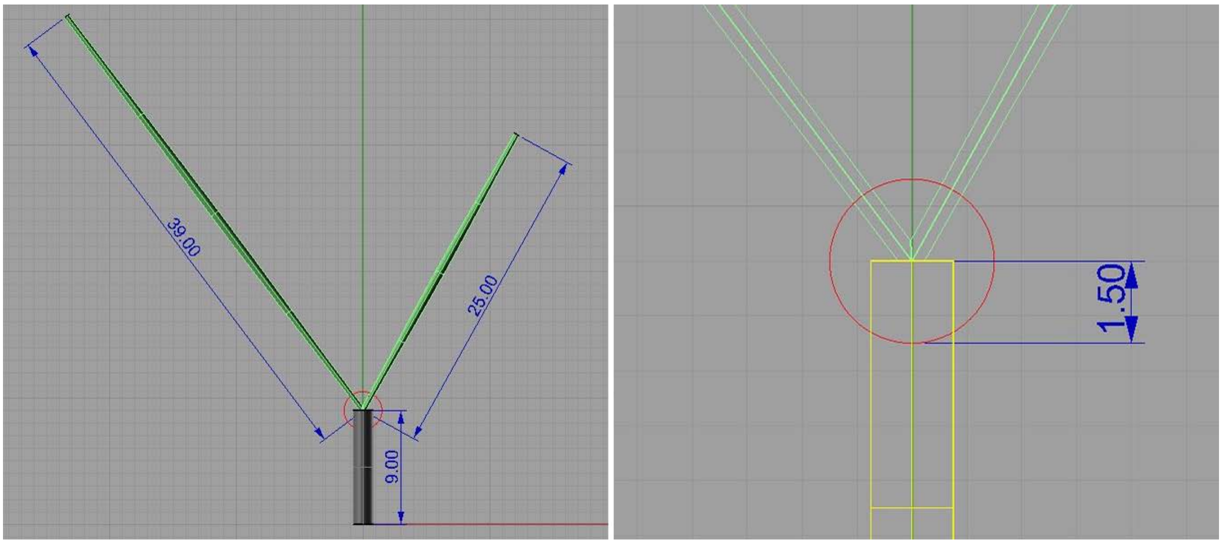


Figure 3.18. Node boundary circle

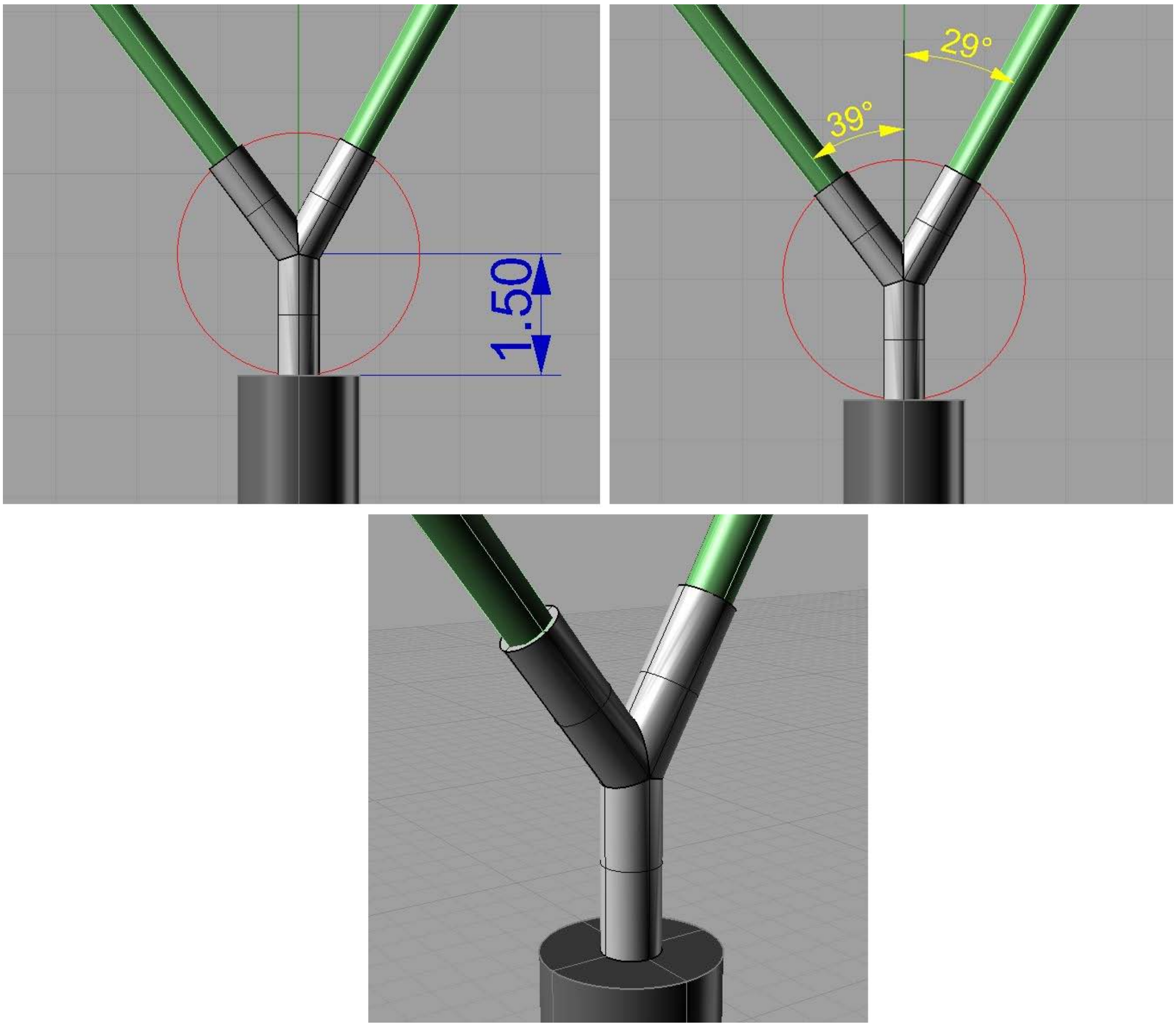


Figure 3.19. Special node arm angles

3.5. Structural Calculations

The structural node in UHCP, designed in Section 3.4.3 has been checked according to the French recommendations for UHPC. The force distribution has been calculated in Section 3.4.

The structural capacity of the structural node is calculated using the equations extracted for the French recommendations for UHPC, presented in Section 3.3.1. Because the dimensions and cross-sectional area is similar for all three node arms, the capacity calculation for one arm will be sufficient and will apply to all node arms.

A. Compressive capacity:

$$f_{cd} = \frac{0,85 * f_{ck}}{\gamma_c}$$

f_{cd} = compressive strength
f_{ck} = characteristic compressive strength
γ_c = material factor=1,3
$f_{ck} = 170$ N/mm^2 (Table 3.4)

$$f_{cd} = \frac{0,85 * 170}{1,3} = 111,1 \text{ N/mm}^2$$

Compressive capacity = 111,1 N/mm^2

B. Direct tensile Capacity

$$f_{ctd} = \frac{0,7 * f_{ctm}}{\gamma_c}$$

f_{ctd} = tensile strength
f_{ctm} = mean tensile strength
γ_c = material factor=1,3
$K = 1,25$
$f_{ctm} = 10$ N/mm^2 (Table 3.4)

$$f_{ctd} = \frac{0,7 * 10}{1,25 * 1,3} = 4,3 \text{ N/mm}^2$$

Tensile capacity = 4,3 N/mm^2

The structural stress checks are presented below. See Figure 3.17 for the denotation of the node arms.

Compressive stress check

Compressive capacity = 111,1 N/mm^2

Compressive stress in arm A = direct compressive stress + bending stress =
7,57 N/mm^2 + 11,32 N/mm^2 = 18,89 N/mm^2 < 111,1 N/mm^2 Satisfied !
Compressive stress in arm B = direct compressive stress + bending stress =
10,06 N/mm^2 + 6,34 N/mm^2 = 16,4 N/mm^2 < 111,1 N/mm^2 Satisfied !
Compressive stress in arm C = direct compressive stress + bending stress =
14,95 N/mm^2 + 4,94 N/mm^2 = 19,9 N/mm^2 < 111, 1 N/mm^2 Satisfied !

Direct tensile stress check

Tensile stress in arm A = bending stress – direct compressive stress =
11,32 N/mm^2 – 7,57 N/mm^2 = 3,75 N/mm^2 < 4,3 N/mm^2 Satisfied !
Tensile stress in arm B = bending stress – direct compressive stress =
6,34 N/mm^2 – 10,06 N/mm^2 = -3,72 N/mm^2 < 4,3 N/mm^2 Satisfied !
Tensile stress in arm C = bending stress – direct compressive stress =
4,94 N/mm^2 – 14,95 N/mm^2 =-10,0 N/mm^2 < 4,3 N/mm^2 Satisfied !

Only the compressive and direct tensile stress have been checked here since these stresses are the defining stresses for the node and the shear forces are negligible. The finite-element-analysis model in *VisionNode* also only extracts the compressive stress and direct tensile stress from the model. These stresses are required for the structural capacity check for brittle materials which is a Mohr-Coulomb-Failure-Check. More information on this subject is presented in Chapter 6.

All structural checks have been satisfied.

3.6. Conclusion

In this chapter a manual design of a structural node in UHPC for the Yas-Hotel building was successfully designed. For this node the UHPC mixture C170/200 has been used.

A geometrical- and structural computer model have been made of the Yas-Hotel building in order to derive a force distribution on the structural node. Using the Eurocode, the multiple loads have been applied to the model and the resulting force distribution have been obtained. With the French recommendations for UHPC, the structural capacity of the structural node in UHPC has been calculated and afterwards structurally checked. All structural checks are satisfied for the manually designed structural node.

The structural node that has been designed in this chapter must not be considered a Special Node because it is a straightforward design of a structural node, does not have a complex 3D-geometry and does not contain geometrical distortions in the form of holes.

Chapter 4.

VisionNode Software Architecture

In this chapter the developed computational structural design tool *VisionNode* is presented and discussed. First of all, the tool is introduced and its functions described. Subsequently, the programming structure of the tool is explained followed by the operational structure of the software. Before finalising the chapter with conclusions, the key features of *VisionNode* are presented.

4.1. VisionNode Software

VisionNode is a computational structural design tool which allows users to design a Special Node in UHPC. Through computational optimisation, *VisionNode* generates optimised designs of structural nodes, while allowing users freedom of design. The optimisation itself aimes at reducing the volume while satisfying predetermined boundary conditions. This means that the optimisation process searches, through numerous iterations, for a specific shape of the structural node that has a minimal volume and at the same time satisfies all boundary conditions. Every iterated solution that complies with the structural boundary conditions, is considered a feasible solution. The best feasible solution of the optimisation process contains the data for the final design of the structural node. This final design is called the Special Node.

Structural optimisation implies that finding the most efficient solution to a structural problem is the goal of the optimisation. Even though this is basically also the case here, the *'most efficient solution'* is defined somewhat differently in this research project. As was explained in the first chapter, optimisation is viewed from a different perspective in this research. The power of computational optimisation is used to find efficient solutions to problems that seem to be illogical and inefficient to begin with. Users of *VisionNode* are allowed to specify design preferences and make design modifications in order to create the Special Node that they want. This is the reason why the tool is called *VisionNode* because the user is able to project a design vision onto the structural node, in spite of it being a computational optimisation tool. For more information on this, the reader is referred to Chapter 1.

4.1.1. Software Overview and basic features

VisionNode operates as a stand-alone windows application where users are presented with a graphical user interface (GUI). The GUI, displayed in Figure 4.1, is the windows that allows interactivity between user and software. In the GUI, users can define parameters such as material properties, geometrical dimensions, connection properties and optimisation algorithm specifications. All design modifications to the structural node are made from this GUI. All the input parameters are discussed in detail in Section 4.4.

The input parameters for *VisionNode*, that are entered through the GUI, consists of:

- mechanical properties for a specific UHPC mixture
- dimensions of the adjacent structural element to which the Special Node is connected
- coordinates of the connection between the adjacent structural elements and Special Node
- transferred external forces from the adjacent structural element to the Special Node
- design modifications and boundary conditions
- Algorithm specific parameters

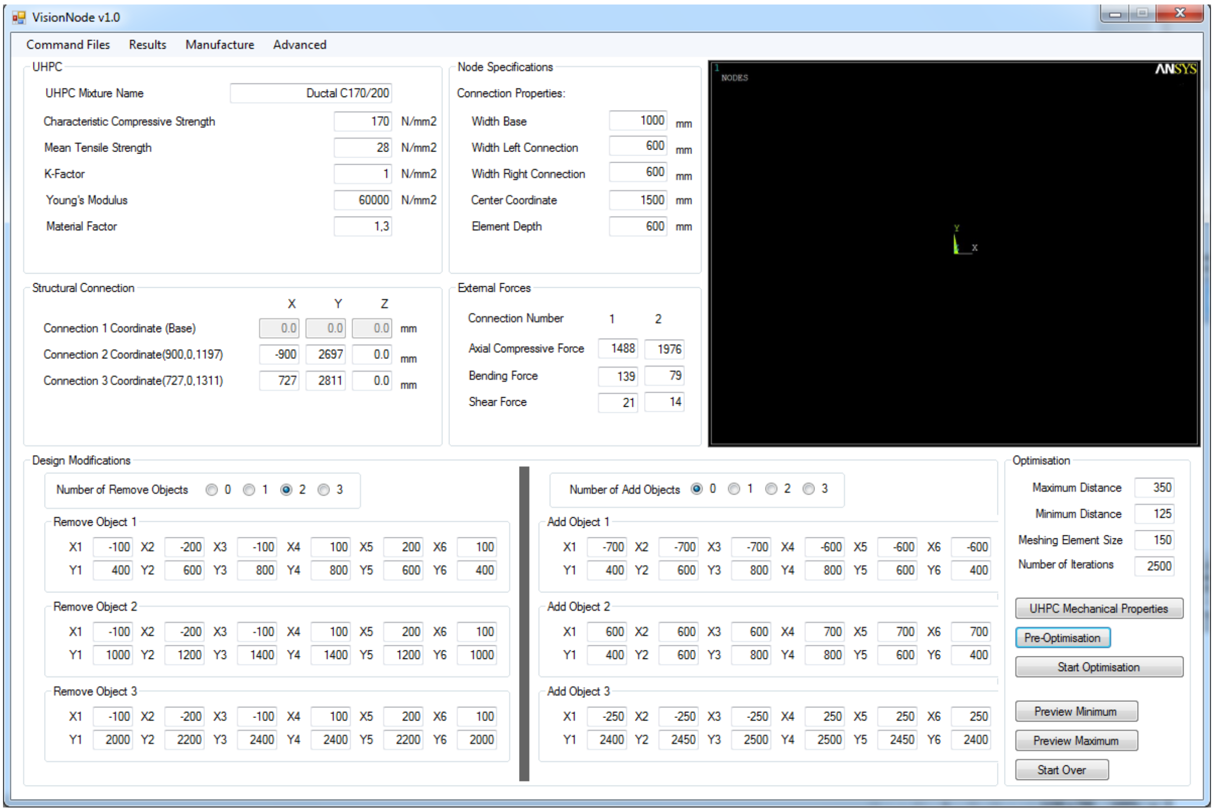


Figure 4.1. VisionNode graphical-user-interface

Any desired UHPC mixture can be input in *VisionNode*. The program is not limited to a single predetermined mixture. This creates more flexibility in the benefit of the design process.

The connection coordinates in the GUI, define the orientation and location of the adjacent structural elements that are connected to the Special Node. The connection properties define the size of the contact surface between Special Node and connecting elements.

The external forces transferred from the adjacent structural element to the Special Node, consist of an axial compression, a bending force and a shear force. The axial force is oriented perpendicular to the plane of the connection and are transferred to the Special Node as a surface pressure load. The direction of the bending and shear forces are controlled by their positive or negative value. More information on the force transfer and distribution can be found in Chapter 6.

The design modifications that can be made through *VisionNode*, consist of the application of Remove-Objects and Add-Objects. Remove-Objects instruct the program that material cannot be used at certain locations in structural node. Add-Objects do the exact opposite and instruct the program that material must be used at certain locations.

After the user has defined all parameters, a preliminary design of a Special Node has been created. *VisionNode* creates all the necessary data for the optimisation process based on this preliminary design. The data consists of the initial node geometry, finite-element-model properties, optimisation algorithm boundary conditions, structural boundary conditions and location of the design-variables that are needed by optimisation algorithm.

Even though *VisionNode* is a stand- alone windows application, it uses several external programs and frameworks in the background for the geometry modelling, finite-element-analysis and optimisation algorithm. For the geometry modelling, *VisionNode* uses the 3D-modelling program Rhinoceros by McNeel www.Rhino3D.com). For the structural analysis of each generated node, *VisionNode* uses the well known finite element modelling program Ansys by Ansys inc. (www.ansys.com. The optimisation algorithm that *VisionNode* uses is called a genetic algorithm (GA). This algorithm is utilized through a optimisation framework called DAKOTA by Sandia International Laboratories (www.sandia.org). The background on the communication between *VisionNode*, Rhinoceros, Ansys and DAKOTA is explained in detail in Section 4.3.

VisionNode generates a number of outputs in the form of textual and graphical files. These files are directly accessible from the GUI after they have been generated. The output consists of:

- Text file: Programming code representing the communication between and the input for the external programs Rhinoceros, Ansys and DAKOTA
- Text file: Output of the finite element analysis
- Text file: Tabular output of the optimisation iteration solutions
- Graphical file: 3D-model of the geometric model of the Special Node
- Graphical file: 3D-model of the finite element analysis model of the Special Node

The content of the output files are discussed in detail in Section 4.4.

In Figure 4.2, a flow-chart is shown which illustrates the basic operation of *VisionNode*.

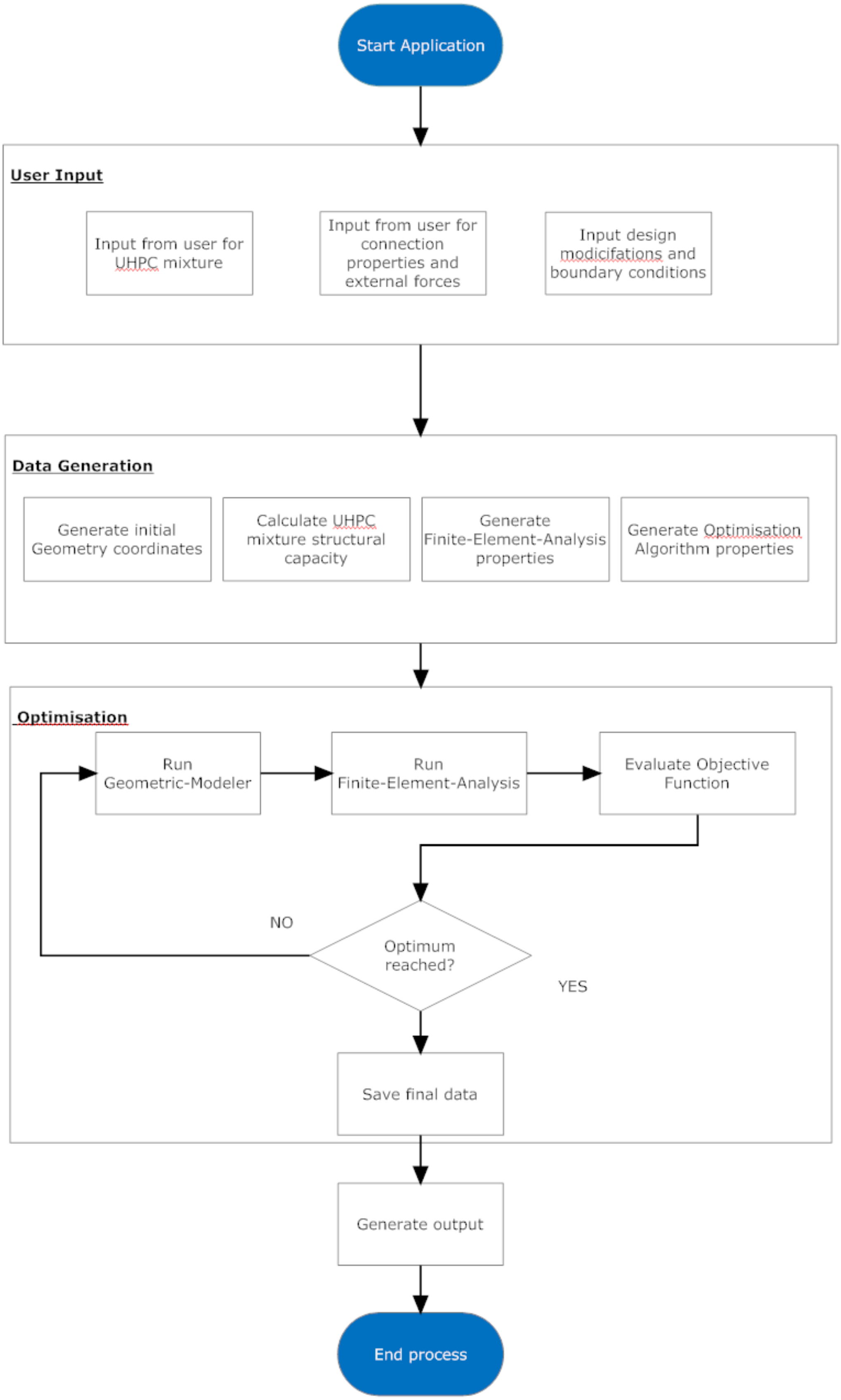


Figure 4.2. VisionNode Operation Flow-Chart

4.1.2. Software objective

The objective of this research project is to determine whether it is technically feasible to create Special Nodes, as defined in this research, in UHPC with fibre reinforcement en without passive reinforcement. In order be able to formulate a conclusion on this, the Special Nodes will have to be created first. For this purpose, it has been chosen to develop a computational structural design tool that creates, analyses and presents Special Nodes. *VisionNode* is this tool and it serves as the means to the objective of this research project. In essence, *VisionNode* is a parametric design tool that creates Special Nodes with the help of an optimisation algorithm. The parameters and design modifications specified by the user, define the characteristics of the Special Nodes.

The objective of *VisionNode* is to accommodate users with freedom of design when designing a structural nodes. This is accomplished by allowing users to specify design preferences and modifications prior to the optimisation process that creates the structural node. The graphical-user-interface provides the means to input the design choices, as described in the previous sub-section. The product of the operation of *VisionNode*, is a Special Node.

Another objective of *VisionNode* is to reduce the time that would normally be needed when designing a Special Node. Manually designing a Special Node would mean creating countless designs and structurally analysing each design until a satisfactory design is found. Since it is difficult to apply straightforward structural rules-of-thumb to a Special Node because of the curved surfaces and holes, a computer analysis for each design would be required. Because of this, the design time would be substantial. What normally would have to be done manually, as described above, *VisionNode* does internally and automatically. Countless designs are created and structurally checked with a finite-element-analysis. Eventually one feasible design with a minimal volume is presented as the final design of the Special Node. This process takes a couple of hours and is much shorter than the manual design and calculation of every design. This is the strength of *VisionNode*.

4.2. VisionNode program structure

4.2.1. Programming Languages

The core-application of *VisionNode* has been written in the programming language known as C-Sharp (C#). C# is an object-oriented programming language developed by Microsoft and is part of their .NET software development platform (Kingsley-Hughes, 2007). Being object-oriented, C# is composed of a collection of individual programming units called classes that interact with each other. This is the way in which a object-oriented programming language operates.

Object-oriented programming is a programming paradigm which presents a set of tools and methods that enable software engineers to build reliable, user friendly, maintainable, well documented, reusable software systems that fulfil the requirements of its users (Budd, 1996). A software system is seen as a community of *objects* that cooperate with each other by passing messages in solving a problem. An object is merely a collection of related information and functionality. It can be something that has a corresponding real-world manifestation, a virtual manifestation, or just some convenient abstraction within a program. The main concepts behind object-oriented programming is the creation of reusable class-libraries that are used in order to create objects. A class-library is and programmable entity that contains variables, methods,

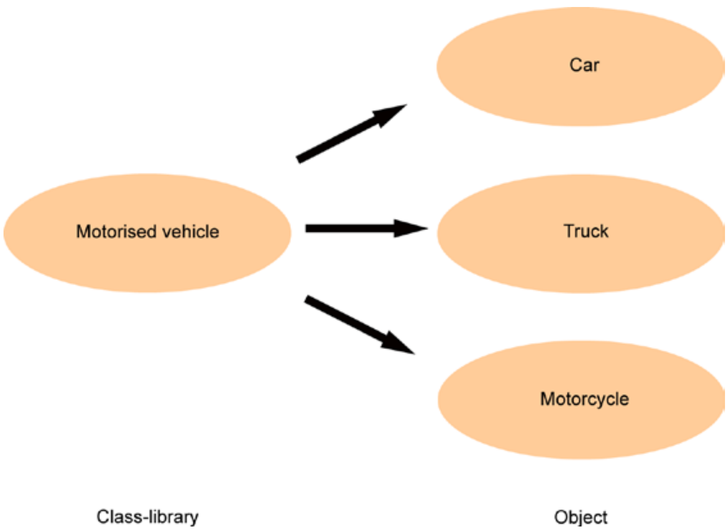


Figure 4.3. From class to object

attributes and other properties for specific subject. By extracting this information from the class-library a specific object can be created that has all the characteristics of the class-library but an identity of its own. This concept is illustrated in Figure 4.3. The motorised vehicle is the class that contains all the properties needed by the object but does not have a specific form. The objects extracts all properties from the class and forms a new entity which a specific form and identity, in this case a car, truck or motorcycle. A class acts as the source and unlimited objects can be made from one class. *VisionNode* also has an extensive class-library form which multiple object are created. This class-library structure is explained in the next sub-section.

Next to the core-application being in C-Sharp, *VisionNode* uses three additional programming languages:

- Visual Basic(VB) developed by Microsoft
- Ansys-Parametric-Design-Language (APDL) developed by Ansys inc
- Perl general purpose Unix-scripting language

VisionNode writes internal *scripts* in VB, APDL and Perl to respectively control and direct the 3D-modelling program Rhinoceros, the finite element modeller Ansys and the optimisation algorithm framework DAKOTA. Scripts are small files containing programming code that allow control of software applications. They are used to facilitate communication between two software application which are written in two different programming languages. For this reason the programming language used in scripts, usually differs from the programming language of the core-application. This is also the case for *VisionNode*.

The scripting language for controlling Rhinoceros is called RhinoScript. It was created by McNeel and is based on Visual Basic. RhinoScript is used by *VisionNode* during the geometric modeling process to generate shape-, form- and section properties for the Special Node.

To control Ansys from within *VisionNode*, the APDL language is used. APDL is based on the programming language Fortran which was originally created for numerical computation and scientific computing. Simple ASCII text-files are used to write the APDL code in order to command Ansys. An ASCII text-file is a text-file in its most basic form without any form of text modifications like bolding, underlining etc. and they are written with the standard texteditor available in Windows. These APDL command-files are created by *VisionNode* and uploaded into Ansys for finite element analysis.

The optimisation algorithm framework DAKOTA, is controlled through a Perl-script. *VisionNode* creates this Perl-script in order to specify how the optimisation must be run. In turn, the optimisation process is directed by the Perl-script which takes care of the smooth interaction between Rhinoceros, Ansys and DAKOTA during optimisation.

4.2.2. VisionNode Class-Structure

In the previous sub-section, it was mentioned that *VisionNode* contains class-libraries, or classes, from which objects are created. The structure between the classes can be illustrated in the form of an so called UML class-diagram. This diagram is displayed in Figure 4.4 and shows how the classes interact with one other. There are a total of 14 classes in *VisionNode*. A description of each is given in the following.

The Structural-Properties-class is one where all information is contained about the structural properties of the node such as:

- The UHPC mixture material properties
- The structural connection properties in the form of the coordinates of the incoming structural elements and the dimensions of the same elements
- External forces that are transferred to from the incoming element to the Special Node

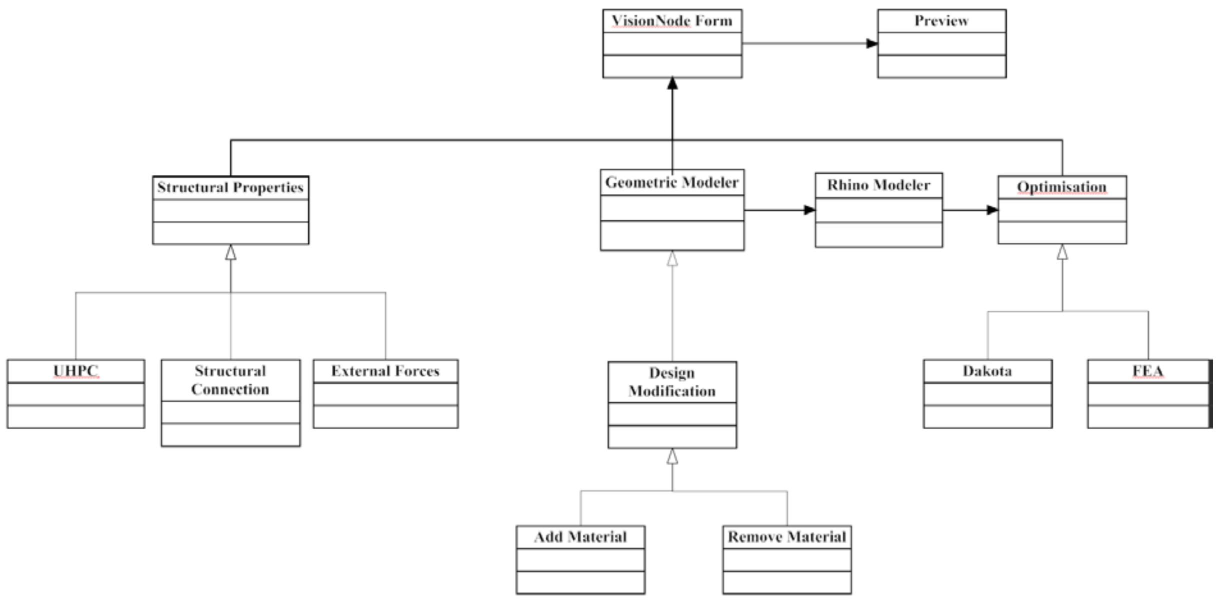


Figure 4.4. UML Class-diagram

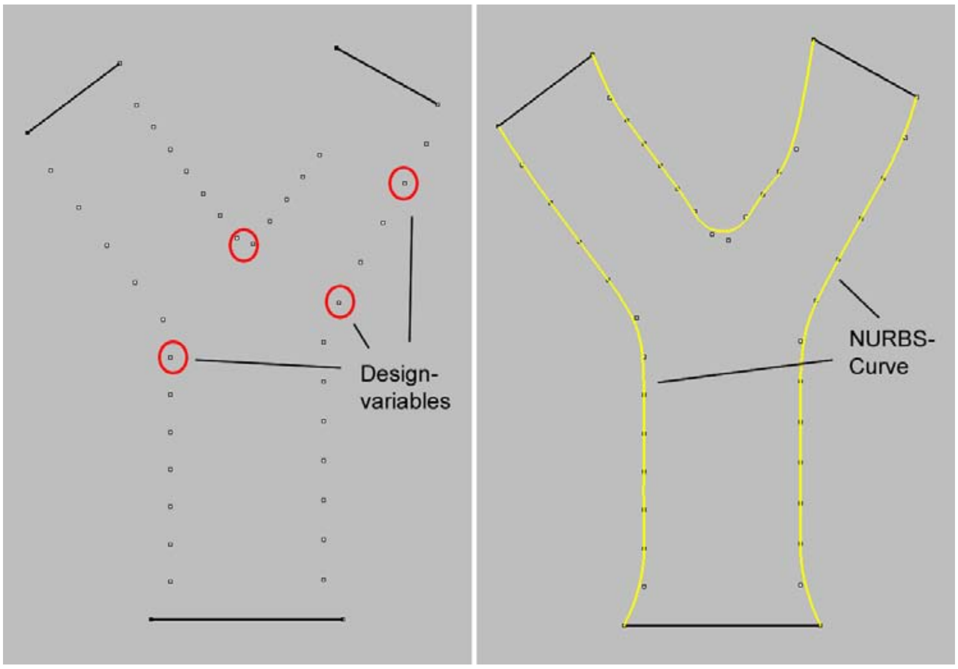


Figure 4.5. Design-variables and the NURBS-Curves created with them

The Geometric-Modeller-class is the largest class in *VisionNode* and contains by far the most code. This class defines the geometry of the Special Node by calculating the X- and Y-coordinates of the design-variables for any given situation. The design-variables are the points in the XY-plane from which NURBS-curves are generated by Rhinoceros. NURBS-curves are Non-Uniform Rational B-Splines and are essentially Bezier-curves based on control points. The design-variables make up the control points for each NURBS-curve. There are a total of 48 design-variables generated by *VisionNode* which are used to create three NURBS-curves. In Figure 4.5 a preliminary design of Special Node is displayed in the form of design-variables and NURBS-curves. The curves, together with the three connection-plane-lines, form the Special Node.

The design-variables are called such because these are the variables that the optimisation algorithm uses to generate a new geometry of the Special Node for each new iteration. The locations of the design-variables are shifted by the algorithm with each new iteration. As a result, a new design for a Special Node is generated for every iteration of the optimisation process. At the end of the optimisation process the best design is presented as the optimum solution.

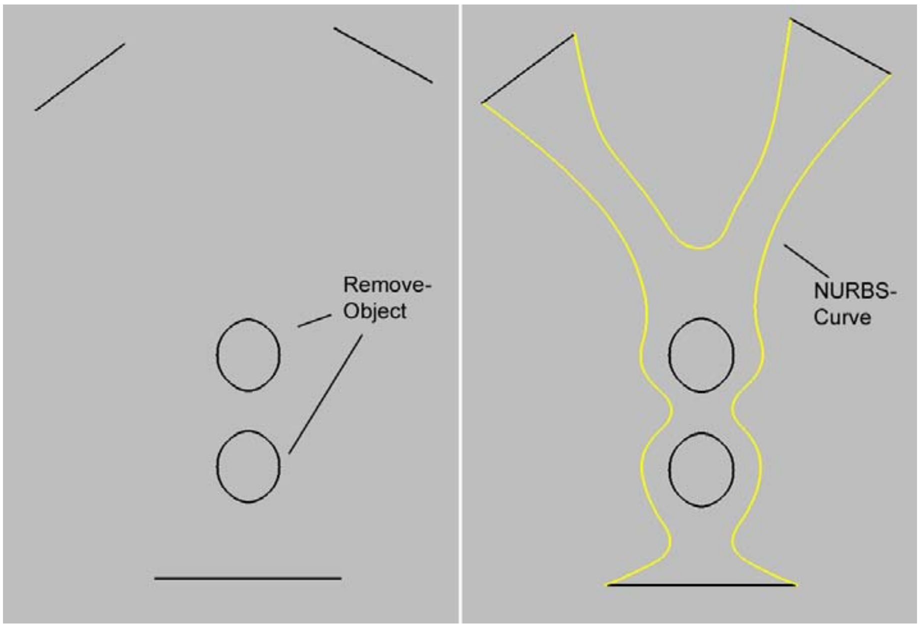


Figure 4.6. NURBS-Curves and the Remove-Objects. The NURBS-Curves are positioned around the Remove-Objects

From the class-diagram in Figure 4.4 it can be seen that the Geometric-Modeller-class gets additional information from the Add-Material-class and the Remove-Material-class. In these two classes the X- and Y-coordinates of the Add- and Remove-Objects are defined. The Geometric-Modeller takes these objects into account and calculated the design-variables such that the Add- and Remove-Objects are incorporated in the Special Node. See Figure 4.6. Note how the NURBS-curves are positioned around the Remove-Objects which are essentially holes in the Special Node.

All the information contained in the Geometric-Modeller-class is fed to the Rhino-Modeller-class which is responsible for creating the NURBS-curves and the connection-plane-lines.

The Optimisation-class contains all the information needed to run the optimisation. In order to do this it needs information from:

- The Rhino-Modeller-class where all NURBS-curves information is stored
- The DAKOTA-class where the optimisation algorithm information and code is stored such as the boundary conditions, number of iterations and the genetic algorithm specific parameters
- The FEA-class which contains all the information and code for the finite-element-model

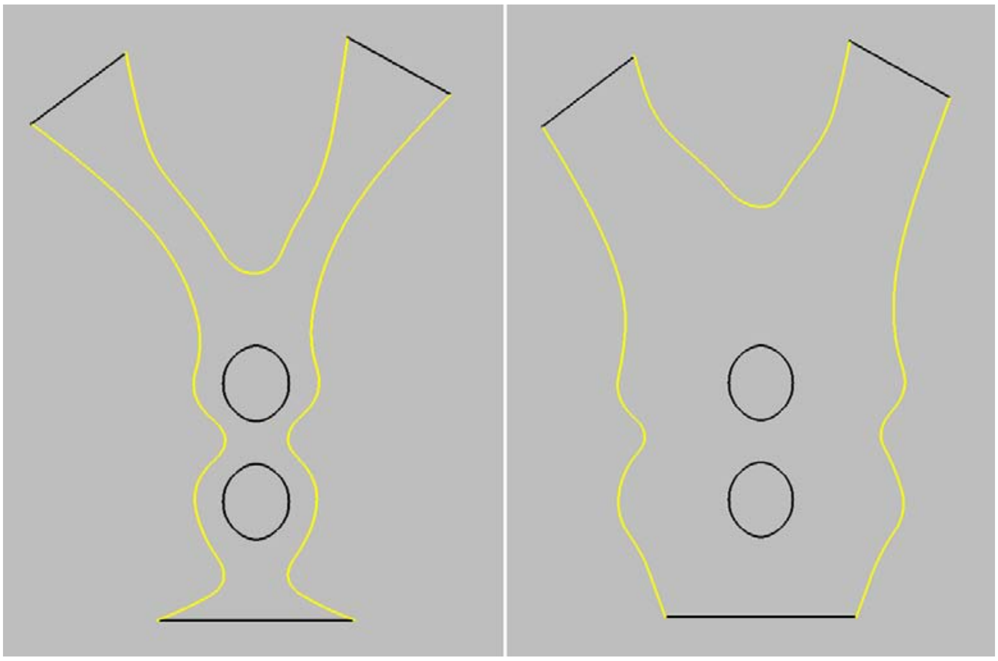


Figure 4.7. Minimum- and maximum geometry of the Special Node

All the previously described classes pass their information to the *VisionNode*-Form-class where the objects are created from the other classes. An interface between the user input parameters and the objects is also created in this class. The *VisionNode*-Form-class can be considered the heart of the program where all actions are taken.

The final class is the Preview-class which contains code for previewing the user created design of the structural node before the optimisation is started. This class was created to provide a method for users to preview the preliminary design and the minimum- and the maximum-geometry of that preliminary design. The preview feature provides the user with feedback on the preliminary design before a lengthy optimisation process is started. If the user is unhappy with the preliminary design, additional modifications can be made and previewed again.

The minimum- and maximum-geometry serves as the geometrical boundary for the optimisation algorithm. During the optimisation the geometry of the Special Node cannot get any smaller than the minimum-geometry as it serves as the lower-geometrical-boundary for the optimisation algorithm. The same applies for the maximum-geometry. See Figure 4.7. This means that the design-variables that are shifted by the optimisation algorithm for each iteration, always stay within the bounds of the minimum- and maximum-geometry.

4.3. VisionNode operational structure

The programming languages, applied scripts and the class-structure of *VisionNode* have been described in the previous section. How *VisionNode* operates and uses all other applications, is explained in this section.

In Figure 4.8 the core of the operation of *VisionNode* is illustrated through a flowchart. Once the operation has been started and the user input parameters have been defined, *VisionNode* generates the initial data and scripts that are needed during the optimisation process. The scripts are needed for communication between the multiple applications and the initial data are the design variables which are created by the geometric-modeller and uploaded to Rhinoceros. After this is complete, *VisionNode* runs the Perl-script to start DAKOTA and run the optimisation process. The Perl-script directs the optimisation process between the different applications.

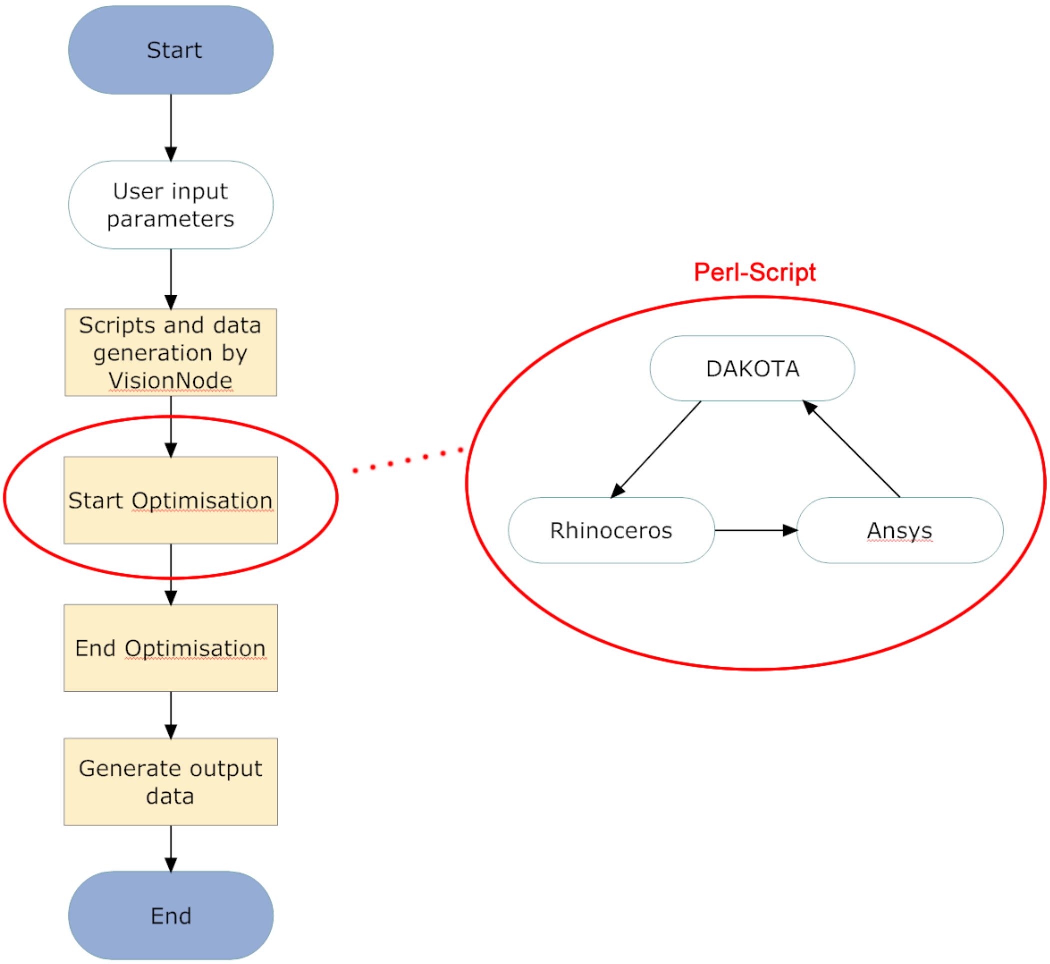


Figure 4.8. Core of the operation of VisionNode

During optimisation DAKOTA, Rhinoceros and Ansys communicate with each other and continuously exchange data. The communication between these applications, is controlled by the Perl-script. This script knows what application has permission to start and what applications has to wait. Since the optimisation process is a data exchange, certain data must be made available first by an application before another application can read the data and start its operation. This process is continued until the optimisation is ended by the Perl-script.

After the optimisation has been initiated, DAKOTA starts the genetic algorithm and creates input data for Rhinoceros. This input data are the new locations of the design-variables which Rhinoceros needs to create the NURBS-curve model. When the NURBS-model of the Special Node geometry has been created, the this model is stored by Rhinoceros as an IGES-model. IGES stands for Initial Graphics Exchange Specification. It is a file format that serves as a neutral data format to exchange digital information among Computer Aided Design (CAD) systems. By storing the Special Node model in the IGES format, the digital geometrical-data on the model, can be read by Ansys.

The next step is the creation of the Finite-Element-Model (FEM). In order to create the FEM, the earlier generated APDL command-file is uploaded into Ansys by the Perl-script. In the APDL command-file, Ansys is instructed to:

- Generate a solid-geometry of the Special Node
- Assign material properties to the model
- Create a finite-element mesh of the solid-geometry
- Assign loads and forces to model
- Run the solver for the structural calculations
- Write the calculation results to a results-file

When Ansys has finished its operation, it shuts down and leaves the results file. Now Perl-script knows that Ansys has ended and recognises the results-file. Subsequently it uploads this file into DAKOTA again, in order to start another iteration-step of the optimisation process.

After the optimum solution has been found, all data is stored and DAKOTA shuts down. The Perl-script knows that DAKOTA has finished and shuts down all other running applications. The optimisation data is stored and the *VisionNode* output data is generated after which

VisionNode stops its operation. The result of this entire operation, is a newly created Special Node. The details of the optimisation algorithm and DAKOTA are discussed in Chapter 5. The details of the FEM and Ansys are discussed in Chapter 6.

4.4. VisionNode key features

The operation of *VisionNode* has been explained in the previous section. In order to make *VisionNode*, and its optimisation process, run correctly and smoothly, a number of features have been implemented in the program. These features are presented and explained in this section.

Interactivity through GUI

VisionNode has been made an interactive application through the implementation of a graphical-user-interface (GUI) which was shown in Figure 4.1. The user can input the following parameters in this GUI:

- Material data for UHPC mixture
  - Characteristic compressive strength in N/mm^2
  - Characteristic tensile strength in N/mm^2
  - Fibre-orientation factor
  - Young's Modulus in N/mm^2
  - Material factor
  - The material capacities are calculated internally by *VisionNode* bases on this data
- Connection coordinates of incoming elements
  - A total of two incoming elements can be connected to the Special Node
  - Designation by X- and Y-coordinate
- Connection Properties
  - Width of the base of the Special Node. This is the contact surface between node and foundation
  - Width of left connection. This is the contact surface between node and left incoming member)
  - Width of right connection
  - Centre-coordinate. This is the coordinate where the node arms cross each other
  - Depth of the node

- External forces transferred from incoming members to Special Node
    - Axial compressive force
    - Bending moment
    - Shear force
- Design modifications
    - Coordinates of Remove-Objects
    - Coordinates of Add-Objects
    - Up to three Remove- and Add-objects can be input
- Optimisation parameters
    - Maximum- and minimum distance
    - Meshing-element size
    - Maximum number of iterations
- Genetic algorithm parameter-set
    - This is done through the Advanced menu-button
    - The GA parameter-set is explained in Chapter 5
- Distance-Factoring

Design Modifications

The design modification are the Remove- and Add-Objects that the user can create. Every Remove- or Add-Object is defined by six points in the XY-plane. Any shape of a Remove- and Add-Object can be created using these six points. In order to create a design-object, the X- and Y-coordinate of each point has to be specified by the user. The coordinates have to be input in a certain way, starting with the first coordinate in the lower left-corner and continuing clockwise with the rest of the points. This is illustrated in Figure 4.9.

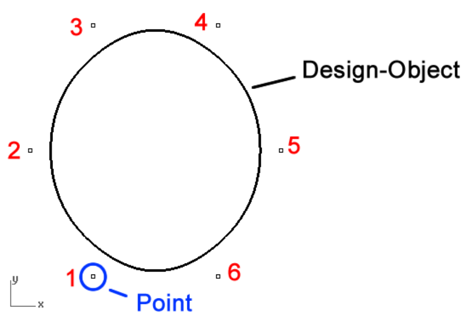


Figure 4.9. Input procedure for Remove- or Add-object

Maximum- and Minimum-Geometry

The maximum- and minimum geometry defines the geometrical boundaries of the design-variables. The created Special Node can never be smaller than the minimum-geometry and it can never be larger than the maximum-geometry. The design-variables can only be relocated by the optimisation algorithm, between these two values. In order words, the maximum- and minimum geometry represent the bandwidth of the design-variables.

Distance-Factoring

The maximum-geometry is created through a feature in *VisionNode* called Distance-Factoring. What this feature does is that the maximum bandwidth of every design-variable, is based on its distance to the Remove-Object. The closer a design-variable is to a Remove-Object, the larger its maximum-geometry becomes. The purpose of Distance-Factoring is to make the optimisation process and results more efficient. High stresses in the node are most likely to occur close to or around the geometrical distortions which are the Remove-Objects. For this reason, extra material is required in the vicinity of the Remove-Objects. However, a design-variable that is far away from a Remove-Object does not need additional material, since the stresses far from the Remove-Object would be lower.

Distance-Factoring is illustrated in Figure 4.10. On the left of this figure, the minimum-geometry of a preliminary design with one Remove-Object is displayed. Next to it is the maximum-geometry with Distance-Factoring. It can be seen that the area around the

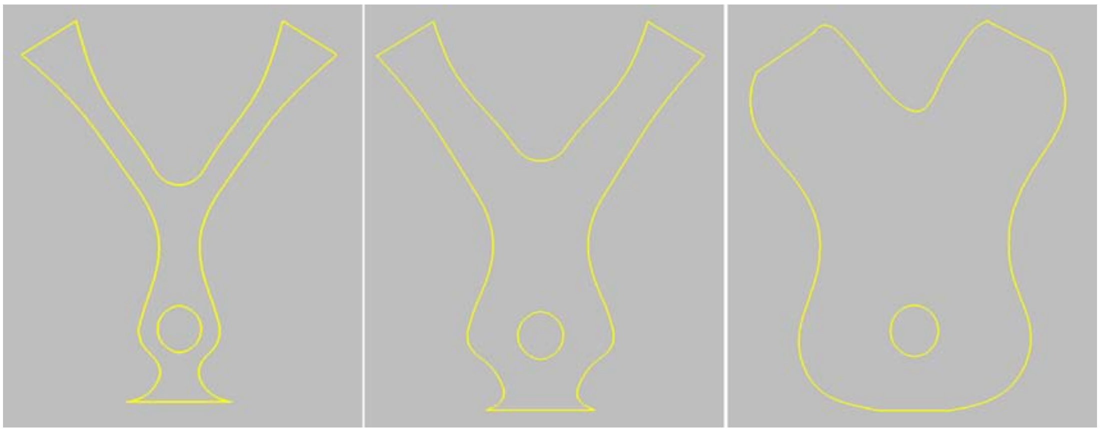


Figure 4.10. Distance-Factoring in VisionNode

Remove-Objects is larger since the maximum-geometry of the design-variables is given a larger value. The farther away a design-variable is located from the Remove-Object, the more it looks like the minimum-geometry since it is given a smaller maximum value. On the right of Figure 4.10 the maximum-geometry of the same preliminary design is shown, but this time Distance-Factoring has been turned off. The results if a maximum-geometry that is equally large for all design-variables. There is a lot of unnecessary material present in this maximum-geometry. It is called unnecessary because high stresses will not occur in the arms of the node, but high stresses will occur near the Remove-Object. This last maximum-geometry is considered inefficient for this reason.

The advantage of Distance-Factoring is that the optimisation algorithm is allowed to focus on the important areas, such as the areas around the Remove-Objects. The result is that the optimisation process takes less time and the end-result looks much better.

It must be noted that the Distance-Factoring is controlled through a factor that can be adjusted at any time by the user. If more material for the maximum-geometry is desired, a higher factor can be input and the maximum-geometry will be adjusted accordingly.

Failure-Catching

It has been explained that the optimisation process is run by DAKOTA and the Perl-script. DAKOTA needs a results-file from the FEM, in order to make the proper adjustments for the next iteration-step in the optimisation process. When Ansys has finished its structural calculations, the results are written to a result-file. However, whenever an error occurs in Ansys, the program will stop functioning and it will exit without completing its calculations. As a result, no result-file is created and DAKOTA will have no results-file to continue the optimisation.

Failure-Catching has been implemented to remedy this problem which is known to occur in Ansys. If Ansys shuts down for some reason without producing a results-file, the Perl-script that oversees the entire optimisation recognised this. In order to let the optimisation continue, the Perl-script creates a 'dummy'-results-file so that the optimisation process can continue.

Furthermore, the dummy-results-file contains data which the optimisation algorithm will use to create a solution of the objective function. The data, however, written in the dummy-file will result in a extremely high value of the objective function. As a result, the algorithm will avoid creating new design-variables that are similar to the one that made Ansys shut down. In other words, the dummy-file fools the algorithm and forces it to generate design-variables that will not cause problems for Ansys. As a result, the optimisation process is made more efficient.

2D-geometry and 3D-geometry switching

The goal of the optimisation within *VisionNode*, is to generate a 3D-geometry of a Special Node with a sculptural quality to it. However, such a 3D-geometry takes a lot of calculation time in the FEM, between 1-10 minutes depending on the complexity of the shape. Because of this, optimisation times up to a day can be expected.

To make the optimisation run more efficiently and take less time to finish, a feature has been implemented in *VisionNode* that switches between a 2D- and a 3D-geometry of the Special Node. In essence, when the solutions of the objective function are far from feasible, *VisionNode* uses a 2D-geometry of the Special Node during the FEM calculations. A 2D-geometry calculation in the FEM takes between 10-30 seconds. This way, the optimisation process can work fast when the solutions have not converged yet. Whenever the solutions become feasible, VisionNode switches to the 3D-geometry and request a 3D-Special Node.

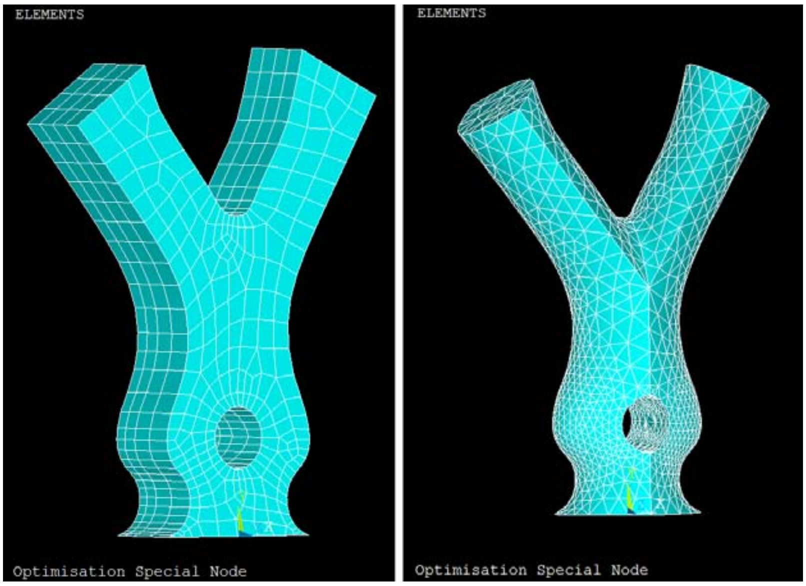


Figure 4.11. Left: 2D-geometry of a Special Node that required little calculation time. Right: 3D-geometry of a Special Node that required a lot of calculation

When considering that during optimisation almost 95% of all the solutions are likely to be infeasible, the advantage of 2D to 3D-switching becomes evident. Because of this feature in *VisionNode*, optimisations take several hours instead of an entire day to complete.

VisionNode Output

After the optimisation has completed and a Special Node is created, *VisionNode* generates:

- Tabular data
- Graphical models
- Ansys Finite-Element-Model of the optimised Special Node
- Rhinoceros model of the same Special node in the form of a *.3dm file

All output data is stored on the C-drive in the output-folder *C:\VisionNodeResults*.

The tabular data from the optimisation process is stored in the *dakota_tabular* file, located in folder *C:\VisionNodeResults\DakotaResults*. The tabular data-file represents all the values of the design-variables, objective function and algorithm-constraint during optimisation. Users can use the tabular data results to see how the optimisation process has evolved.

The graphical models are meant to see what the end result of the optimisation and what the Special Node looks like.

The FEM model of the Special Node can be used to evaluate the calculation results, to see the stress distribution through the node and other structural data. It can be useful during manufacture and casting of the Special Node in UHPC.

The Rhinoceros model can be used to modify the Special Node, when this is desired. The *.3dm file in which the Rhinoceros model is stored, can be used for the manufacture of the Special Node.

4.5. VisionNode application prerequisites

In order for the current version of *VisionNode* to work on a Windows computer, the following is required:

- The *VisionNode* application itself
- Ansys 12.0 installed
- Rhinoceros 4.0 installed
- DAKOTA installed
- Strawberry-Perl installed

Additionally, the following steps have to be taken in order for *VisionNode* to function:

- The location on the harddrive of the Ansys, Rhinoceros, DAKOTA, Strawberry-Perl executables, **must** be assigned to the *Windows-path*. This is done through the *Environmental-settings* of the Windows operating-system.
- Windows operating-system must be installed on the C:\ drive of the computer

When all of the above has been completed, *VisionNode* should run without any problems.

4.6. Conclusions

In this chapter, the computational structural design tool *VisionNode* has been presented and all its functions explained. All the discussed operational aspects of the tool have been successfully implemented in VisionNode and work according to expectations.

Using the GUI of VisionNode, any desired Special Node including all design preferences and modifications, can be created. The GUI has been found to function correctly and accoring to the users wishes.

The optimisation process in VisionNode is controlled by the Perl-script and functions correctly. All separate applications such as Rhinoceros, Ansys and DAKOTA are utilised correctly and the optimisation process performs to expectation.

The key features discussed in Section 4.4, that have been implemented in VisionNode, make the tool much more responsive and easy to use. The operation of the tool and the optimisation process have been greatly improved with the implementation of the key features.

Chapter 5.

Genetic Algorithm Structural Optimisation

In the previous chapters it has been explained that the computational structural design tool uses computational optimisation to create Special Nodes. In this chapter the background on the optimisation process is explained in detail. Firstly, the genetic algorithm and its characteristics are discussed followed by a section on the optimisation framework DAKOTA which supplies the genetic algorithm. Subsequently, the implementation and the operation of DAKOTA and its algorithm in *VisionNode* is explained. The chapter is finalised with conclusions.

5.1. Genetic Algorithm

The structural optimisation process within *VisionNode* is powered by a optimisation algorithm called a genetic algorithm (GA). GA belongs to the family of evolutionary algorithms which are algorithms that try to mimic the evolutionary process of natural selection, reproduction, mutation and survival of the fittest. The GA is a mathematical stochastic search method that was first devised by Holland (Holland,1975) and later extended by Goldberg (Goldberg,1989).

Though a continuous iteration process, the algorithm tries to find the optimum feasible solution to a given problem.

Before the GA starts its search, it is first supplied with an objective function, a set of design-variables and algorithm constraints. The objective function is the function that the GA is trying to find the best feasible solution for. This usually consists of finding the minimum value of the objective function in the search space. A search space if the space in which all the feasible solutions are located. A solution is considered feasible when it satisfies the predetermined algorithm constraints. When a solution does not satisfy the constraint, it is considered infeasible and is omitted from the search space. The GA searches for solutions by changing the values of the design-variables with each iterations and checking the outcome of the corresponding objective function. By repeating this process continuously, the algorithm searches for the best possible feasible solution which is called the optimum solution. The operation of the GA is explained in detail in Section 5.1.1.

The power of GA's lies in the fact that they are robust techniques and can deal successfully with a wide range of problem areas, including structural engineering problems. They are also characterised by the fact that they search for a global optimum of a given problem and avoid local optima. Because of this they are very useful for global optimisation problems such as the optimisation problem in this research project. A global optimum means that the algorithm searches for a design-variable set which represents the overall optimum solution of a problem. Local optima are solutions that represent a localised optimum in the solution-range and usually do not represent the best solution for the problem as a whole.

The GA that *VisionNode* uses, is supplied by an optimisation framework called DAKOTA. This framework contains many optimisation algorithms and GA is one of them. *VisionNode* interfaces with DAKOTA to use the GA and run the optimisation process. More information on this can be found in Section 5.2.

5.1.1. Operation of the Genetic Algorithm

In has been stated earlier that the GA is an evolutionary algorithm that tries to imitate the evolutionary process. How this process operates from begin to end is illustrated in Figure 5.1.

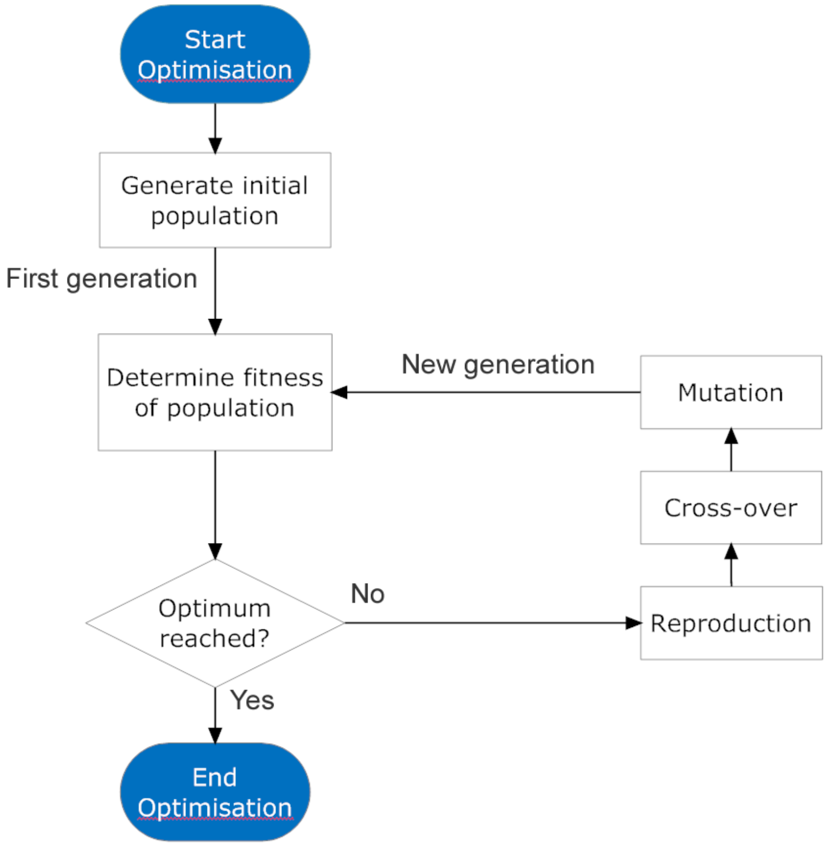


Figure 5.1. Genetic algorithm optimisation process

After the optimisation process has been initiated, the algorithm start by generating an initial random population of *n* chromosomes. The design-variables are encoded as binary-strings which in turn represent the chromosomes. There are other methods of encoding available for a GA, as a string of integers for example, but in this research project the chromosomes have a binary encoding such as:

0100111010101001010101 (one chromosome)
1101000011101010110111 (another chromosome)
.....

An important aspect of a GA is the initial population size which must be set manually. Choosing the appropriate size for the initial population is therefore also important. When the population size is too small, there is little genetic material available when creating a new generation during the optimisation. As a result, diversity among generations is minimised and the difference between the solutions is also minimised. The consequence of this can be that the algorithm converges to a local optimum instead of a global optimum. When the initial population size is too large, however, there is a lot of genetic material available. This maximises the diversity between the subsequent generations and solutions. Even though this seems like a advantage, the disadvantage of a very large population size is that the algorithm will converge very slowly as there is more data to consider before converging. The goal is to find a minimum initial population size which has the required diversity but will not increase the optimisation time unnecessarily.

After the initial population of chromosomes has been created, the fitness of that generation is determined. The fitness of a generation is related to the objective function. If the generations improves the value of the objective function, than it is given a higher fitness score than a generation that degrades the value of the objective function. In other words, the more a generation improves the objective function, the higher the fitness score it receives will be.

After the fitness calculation, the algorithm checks if an optimum has been reached. This is never the case for a first generation since the algorithm needs more solutions for comparison. For this reason, the next step in the optimisation process is the reproduction. During reproduction, the existing population members are labelled as the parents. Reproduction occurs by cross-over where individual parents, or chromosomes, are recombined to create offspring, new chromosomes, which will comprise the new generation. After the cross-over has taken place, mutation occurs. Mutation means that a small part of the newly created chromosome is changed randomly in order to create offspring that is not an exact copy of its parents. This is also true in nature, where children are never exact copies of their parents. The process of reproduction, cross-over and mutation is displayed in Figure 5.2.

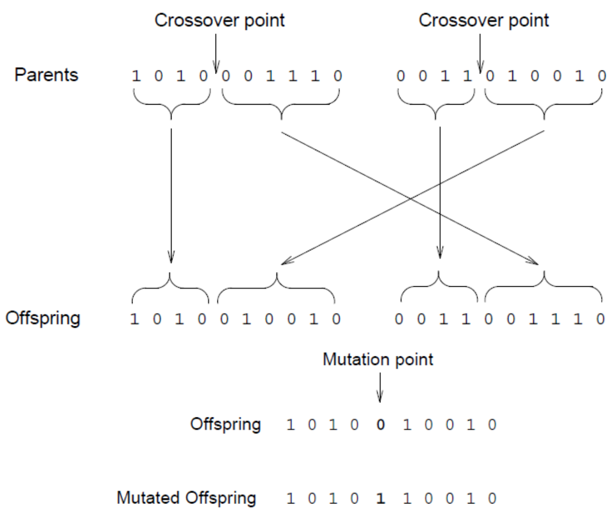


Figure 5.2. Reproduction, cross-over and mutation in a GA (Beasley et al., 1993)

The cross-over probability determines how often cross-over occurs during reproduction. The cross-over probability is expressed through the cross-over-rate which has a range of 0.0 - 1.0. A value of 0.5 indicates a cross-over probability of 50 % which means that during reproduction half of the offspring are created through cross-over and half are exact copies. A value of 0.0 would mean that no cross-over occurs and all the offspring are exact copies of the parents. This is never a good idea because diversity in the generations, and consequently the solutions, is lost. A cross-over rate of 1.0 would mean that all offspring are created through cross-over. This is not a good idea either because the GA would turn into a random searcher since all the chromosomes, and all possible good solutions, of the previous generation are lost. A suitable value for the cross-over rate for any optimisation problem, must be determined through testing.

The mutation probability determines how often mutation occurs and is expressed through the mutation rate which has a range of 0.0-1.0. A mutation rate of 0.5 indicates a mutation probability of 50 % which means that half of the offspring undergoes mutation to their chromosomes. By adjusting the mutation rate, the diversity of new population can be influenced. A high mutation rate has high diversity as a result, but risk the GA turning into a random searcher. A low mutation rate has low diversity as a result and is always advised. Like the cross-over rate, the mutation rate is meant as a control mechanism to direct the GA during optimisation. A suitable value for it must also be determined through testing.

The product of reproduction, cross-over and mutation, is a new generation of population members. For this new generation, the fitness is determined and the score evaluated by the algorithm. This process is repeated over and over again until the solution converges to an optimum. During the optimisation process, the fitness of all generations is compared to one other in order to make a convergence of the solutions possible. The algorithm changes the design-variables, and thus the solutions, in the direction of the generations with high fitness. This way convergence can occur and the algorithm can find an optimum solution.

In this section the initial population size, cross-over rate and mutation rate have been discussed. These are parameters to control how a GA performs and how efficiently it perform. Next to these three parameters, there are more parameters available through which the GA can be controlled. All the parameters together are called the GA parameter-set and are explained in the Section 5.2.

5.2. DAKOTA

DAKOTA (Design Analysis Kit for Optimisation and Terascale Applications) is a open-source framework that provides a flexible problem-solving environment for design and performance analysis of computational models. Developed by Sandia International Laboratories, the framework contains, among others, algorithms for optimisations with gradient- and non-gradient-based methods for optimisation (DAKOTA users-manual, 2009). Being an open-source project, DAKOTA is freely available world-wide. The GA from DAKOTA that is applied for this research project, is an example of a non-gradient-based optimisation method. The version of DAKOTA that has been used in this research project is version 5.0.

In this section the way in which DAKOTA operates is explained. Additionally, the GA and its properties contained within DAKOTA are described in detail. How DAKOTA has been implemented in *VisionNode*, is discussed in Section 5.3.

5.2.1. Operation of DAKOTA

The basic method in which DAKOTA operates is illustrated in Figure 5.3. DAKOTA is utilised by interfacing it with a simulation code, or simulator, provided by the user of the framework. This simulation code can consist of an application or script which receives input parameters from DAKOTA and delivers output parameters in the form of a result file back to DAKOTA. This result file is then processed by DAKOTA and a new parameter file is created which is forwarded to the simulator to initiate a new iteration-cycle. The simulator in this research project is *VisionNode* itself. The interface between DAKOTA and the simulator is created and controlled by a Perl-script which was explained in Chapter 4.

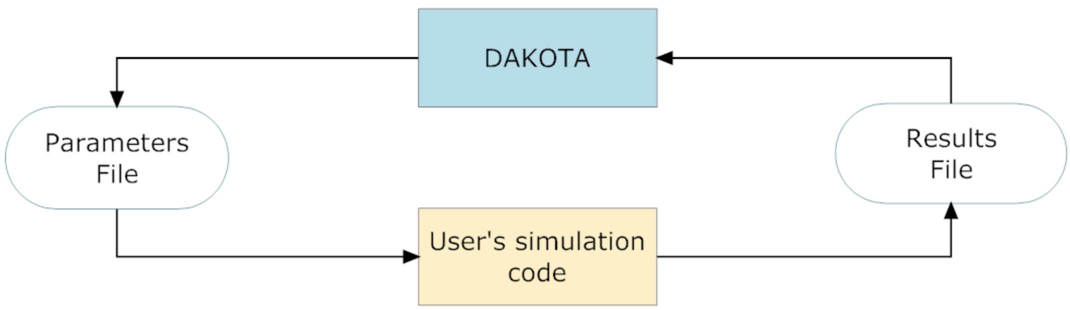


Figure 5.3. DAKOTA method of operation

The operation is basically a data exchange between DAKOTA and the simulator by reading and writing short data files. Because of this, advanced programming skills are not required to operate DAKOTA. The user is only required to know how to read and write the parameters and results file from within the simulator.

DAKOTA is executed though a DAKOTA-input file. In this file, users can specify what type of analysis or algorithm is requested and what the names and locations of the parameters- and result-files associated with the simulator are. During its operation, DAKOTA automatically executes the user's simulation code by creating a separate process external to DAKOTA. Before *VisionNode* runs DAKOTA to utilise the GA, it creates all files necessary for its operation. The content of these files and the basis of the interface between *VisionNode* and DAKOTA is explained in Section 5.3.

5.2.2. Genetic Algorithm in DAKOTA

Version 5.0 of DAKOTA currently provides several variants of single- and multi-objective evolutionary algorithms. The algorithm used in this research project is the single objective genetic algorithm or SOGA as it is designated by DAKOTA. The SOGA method is a global optimisation method contained in the JEGA-library of DAKOTA and was written by John Eddy in 2001.

The SOGA method supports general constraints and a mixture of real and discrete design-variables. The amount and designation of the design-variables are supplied to DAKOTA by the users simulator. Furthermore, the upper- and lower boundary of the design-variables also have to be supplied, so that DAKOTA knows between what values the design-variables have to stay.

Since the SOGA method is a single objective algorithm, a single objective function has to be provide by the user. In DAKOTA, the algorithm is always trying to minimise the value of the objective function. If maximisation is desired instead of minimisation, the objective function must be multiplied with minus-one.

SOGA also accepts any number of nonlinear-equality constraints and nonlinear-inequality constraints. The nonlinear-equality constraints are used to indicate a target value which the optimisation algorithm must try to reach. This target value is by default set to zero but can be assigned any desired value. A mathematical expression for the nonlinear-equality constraints where $g(x)$ is the constraint-function:

$$g(x) = g_t$$

$$g_t = \text{target-value : 0.0} \quad \text{(default)}$$

The nonlinear-inequality constraints in DAKOTA are used to specify between what values a solution is considered feasible. For this reason, the user has to provide an upper- and lower boundary for the constraints. The mathematical expression of the nonlinear-inequality constraint where $g(x)$ is the constraint-function:

$$g_l \leq g(x) \leq g_u$$

$$g_l = \text{lower-boundary}$$
$$g_u = \text{upper-boundary}$$

Next to the design-variables, objective-function and constraints, there are several algorithm specific parameters which are used to control and direct the GA with. These parameters are called the GA parameter-set:

- Seed
- Initial population size
- Initialisation type
- Cross-over type
- Cross-over rate
- Mutation type
- Mutation rate
- Fitness type
- Replacement type
- Convergence type
- Percent change
- Number of parents
- Number of offspring
- Maximum iterations
- Maximum generations

The initial population size, cross-over rate and mutation rate have been explained in the previous section. The others parameters are explained in the following. Note that for every GA parameter in DAKOTA there are several options. All the available options are note discussed in this chapter. However, the options selected for *VisionNode* are discussed in Chapter 7 where the results of this research project are presented.

The *seed* in DAKOTA is a random string of integers that is used as a mechanism for making a stochastic optimisation repeatable. Though the string of integers is random by default, the user has the option to input a specific seed which will generate identical optimisation results for every optimisation run. When the seed remains a random number, optimisation results will always be different for subsequent optimisation runs.

The *initialisation type* defines the type of initialisation for the GA. It is used to generate the first solution set for a certain initial population.

The *cross-over type* and *mutation type* define what method is used for cross-over and mutation. Note that these parameters are different from the cross-over and mutation rate discussed in Section 5.1.

The *fitness type* determines how the fitness of a population is calculated and how strongly differences in fitness are weighted during the process of selecting parents for cross-over.

The *replacement type* determines how current populations and newly generated offspring are combined to create a new population. This control is mainly used not to lose good solutions during subsequent iterations.

The *convergence type* in DAKOTA defines the convergence criteria for the algorithm. It is used to stop the algorithm when the newly generated solutions do not improve the objective function any more. The *percent-change* parameter determines how much variation is allowed between solutions when deciding that the algorithm is converging. When the difference between solutions drops below the percent-change for a number of subsequent iterations, the algorithm is stopped.

The *number of parents and offspring* are parameters that define how many parents are taken during cross-over to generate the desired number of offspring. This is used to control the diversity in subsequent populations by indicating from how many genetic material the new offspring is created.

The *maximum iterations and generations* parameters are used to manually stop the GA.

Now that the GA, DAKOTA and their properties have been explained, how they are used in *VisionNode* is explained in the next section.

5.3. Implementation in VisionNode

The implementation of DAKOTA, and thus the GA, is accomplished through an interface between several components of *VisionNode*. The interface is established through a number of script-files that are needed for the communication between the *VisionNode* components. The details of the interface and the scripts are explained in Chapter 4. For the interface to utilise DAKOTA correctly, first a DAKOTA-input file is created by *VisionNode* which contains, among others, the following:

- GA design-variables
- GA objective function
- GA constraints
- GA parameter-set
- Names and location of parameters-file and results-file

A total of 48 design-variables are used during optimisation in *VisionNode*. They represent coordinates in the XY-plane and are needed to create the geometry of the Special Node for every iteration. Aside from the amount and designation of the design-variables, DAKOTA also needs the upper- and lower-boundary of the design-variables. The minimum- and maximum geometry of the Special Node, which is explained in Chapter 4 Section bla..., determine the values of the upper- and lower-boundary of the design-variables.

The objective function for the optimisation, is the volume of the Special Node. During its operation, the algorithm is trying find solutions for the objective function that minimise its value and consequently the volume of the Special Node. The solutions have to be feasible to be accepted by the algorithm as such. A solution that satisfies the algorithm constrain is considered feasible.

The constraints for the optimisation in this research project consists of a structural failure-check in the form of a Mohr-Coulomb-Failure-Check, which is explained in Chapter 6. The constraint is input as a nonlinear-inequality constraint which determines an upper- and lower-boundary between which the objective function is feasible. The nonlinear-inequality constraint in *VisionNode* is:

$$g_l \leq g(x) \leq g_u$$

$$g(x) = \text{Mohr-Coulomb-Failure-Check}$$
$$g_u = \text{upper-boundary: } 1{,}0$$
$$g_l = \text{lower-boundary: } 0{,}6$$

The lower-boundary of the constraint is chosen 0,6 and not 0,0 in order to force the algorithm to create efficient Special Nodes that do not contain redundant material. For example, a Special Node with a Mohr-Coulomb-Failure-Check of 0,1 would be structurally feasible, but would contain too much unnecessary material. As the goal of the optimisation is to minimise the volume, it is pushed in that direction by setting the lower-boundary to 0,6 and not lower.

The parameters of the GA parameter-set have been explained in the previous section. The chosen parameters are presented in Chapter 7.

As was stated earlier, the optimisation process is a data exchange between DAKOTA and other components of *VisionNode*. DAKOTA supplies the input parameters-file that is needed by the geometric-modeller. Which in turn creates the model for the Special Node and sends it to the Finite-Element-Model (FEM) to do the structural calculations including the Mohr-Coulomb-Failure-Check for that specific geometry. The FEM than creates a data file which contains the results of the calculations. The result-file is uploaded to DAKOTA which passes the results to the GA. Finally, the GA performs its operations based on the result-file and asks DAKOTA to generate the new parameters-file. This cycle is one iteration in the optimisation process and continues until convergence of the solutions occurs.

5.4. Conclusions

In this chapter, the optimisation framework DAKOTA and the genetic algorithm implemented in *VisionNode* has been explained. DAKOTA and its genetic algorithm, have been successfully implemented in *VisionNode* to perform optimisations.

VisionNode provides design-variables, an objective function, algorithm constraints and specific algorithm parameters to the genetic algorithm and oversees the optimisation process during its operation. The design-variables are points in the XY-plane, the objective function in *VisionNode* is the volume of the Special Node and the algorithm constraints consists of a Mohr-Coulomb-Failure-Check. All these parameters are successfully assigned by *VisionNode* to the genetic algorithm.

DAKOTA and the genetic algorithm have been found to function properly and correctly in *VisionNode*.

Chapter 6.

Finite Element Model

In the previous chapters is has been explained that the optimisation process of *VisionNode* starts by creating a geometric model of the Special Node which is based on input parameters that are supplied by DAKOTA. Every iteration-cycle of this optimisation process, starts by DAKOTA creating an parameter file that is used to create the Special Node geometry. This is followed by the structural calculation of that specific geometry which produces a result-file that contains all the structural data of that specific Special Node. DAKOTA needs this results-file to end the current iteration-cycle and start a new one. The structural calculations, that produce the result-file, are conducted through a Finite-Element-Model (FEM).

In the first section of this chapter the properties and operation of this FEM are explained. The characteristics of the implementation of the FEM in *VisionNode* is discussed Section 6.3. The chapter is finalised with conclusions.

6.1. Finite–Element–Model properties

In this research project, the FEM-software package Ansys by Ansys Inc. is used for the finite-element-analysis. Before the calculations with a model can be initiated, there are an number of properties of a FEM that have to be defined first. These properties define the model and how it performs during the solution process. The properties are:

- Finite-element type
- Material properties
- Geometry of the structural element
- Finite-element meshing type
- Finite-element meshing size
- Loads on the model
- Boundary conditions

The finite-element type describes the element that is used to make up the modelled structural element. Every element type has its own specific properties, advantages and disadvantages. For every FEM the choice of the element-type depends, among others, on factors such as:

- what are the real life structural properties that the model has to simulate
- what type of material is used
- what material behaviour is desired
- what specific calculation results are needed

Ansys contains an extensive library of element-types from which two have been chosen for this research project. The first element-type is called a SOLID95-element and is presented in Figure 6.1.

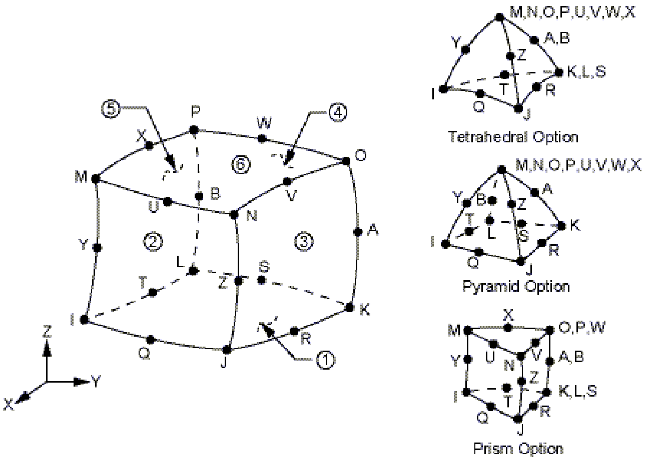


Figure 6.1. SOLID95 element (Ansys Theory, 2007)

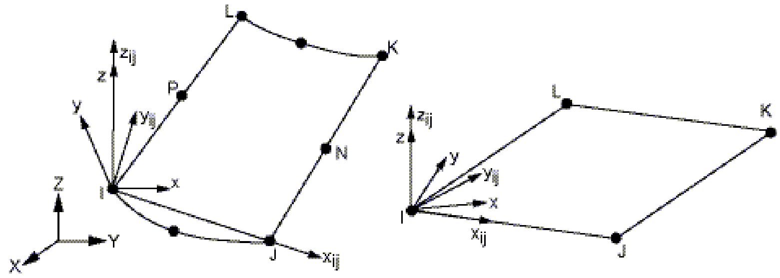


Figure 6.2. SURF154 element (Ansys Theory, 2007)

The SOLID95 element is a 20-node element with three degrees of freedom per node and is suitable for geometries with irregular shapes and curved surfaces. The element has elasticity, plasticity, creep, shrinkage, large deflections, large strains, stress stiffening and other special features. It can process surface pressure loads and concentrated point loads and it is suitable for concrete structural elements.

The second element-type used in this research project is called a SURF154-element and is displayed in Figure 6.2. The SURF154-element is a 8-node surface element that is used for various load and surface effect applications. It is applied to a surface of a 3D-object after which the loads of surface effects are specified. This element is used for applying shear loads to the Special Node. Because the SOLID95-element can only take surface loads perpendicular to the surface plane, it cannot handle shear forces that are parallel to the surface plane. This is the reason why the SURF154 element has been implemented in the FEM of *VisionNode*. The element itself has no mass and is only used to overlay loads and surface effects. It can process surface pressure loads in every direction.

The material properties in the FEM are obviously related to UHPC and consist of a linear material model to which a Young's-modulus, Poisson-ratio and a material density are supplied. Since any type of UHPC mixture can be input in *VisionNode*, the values of these properties are not fixed. In the case of the UHPC mixture Ductal, the material properties are:

- Young's Modulus= 60000 N/mm^2
- Poisson-ratio = 0.2
- Material density = 2500 kg/m^3

The geometry of the Special Node is obtained from the geometric-modeller and Rhinoceros. The geometric-modeller calculates the location of the design-variables which consist of coordinates in the XY-plane. After uploading the design-variables into Rhinoceros, NURBS-curves are created from the design-variables. All the NURBS-curves from Rhinoceros are exported to Ansys which create area's and solid based on these lines. *VisionNode* creates two different geometries of a Special Node, a 2D-geometry and a 3D-geometry. Each of the geometries are used under different circumstances by *VisionNode*. The two geometries and their properties are explained in detail in the Section 6.2.

The finite-element meshing type is property that defines the accuracy of the solutions generated for the FEM. Furthermore, certain properties can be assigned to the meshing element type which influence the chance of a successful mesh-creation of the FEM. There are two meshing types used by *VisionNode*. The first is a hexahedral-element, or brick-element, and can be described as a rectangular block. The second finite-element meshing type is a tetrahedral-element and can be described as a pyramid. The hexahedral-element is used for the 2D-geometry and the tetrahedral-element is used for the 3D-geometry of the Special Node.

Aside from the meshing type, the way in which the mesh is created is determined by the property called Mesh Key in Ansys and consists of the free-meshing and mapped-meshing option. See Figure 6.3. The difference between free – and mapped meshing of a finite-element is that when a mapped-mesh is used, the meshed elements are constricted to a certain pattern and shape. When a free-mesh is applied, the meshed elements are not constricted to a certain pattern and shape. The benefits of free-meshing is that meshed can be created for complex shapes since there is no restriction in pattern or shape. Meshing a complex shape with mapped-meshing is more difficult and the chance of a meshing failure is increased. The trade-off is that mapped-meshed elements require less calculation time than free-meshed elements. In *VisionNode*, free meshing is used because the Special Node have a complex 3D-geometry. This increases calculation time but minimises the chance of a failed mesh generation.

The meshing element size is a property that determines the size of the largest meshed element in the model. The rule of thumb here is, the smaller the meshing size, the more accurate the solutions. Again, the trade-off is an increase in calculation time. Ansys offers an additional feature that allows the adjusting of the element size based on the complexity of the geometry of the modelled element. This feature is called SmartSize and it is used to let Ansys decide where

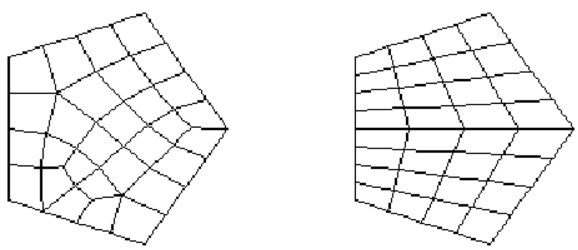


Figure 6.3. Meshing keys. Left: free-meshing. Right: mapped meshing (Ansys Theory, 2007)

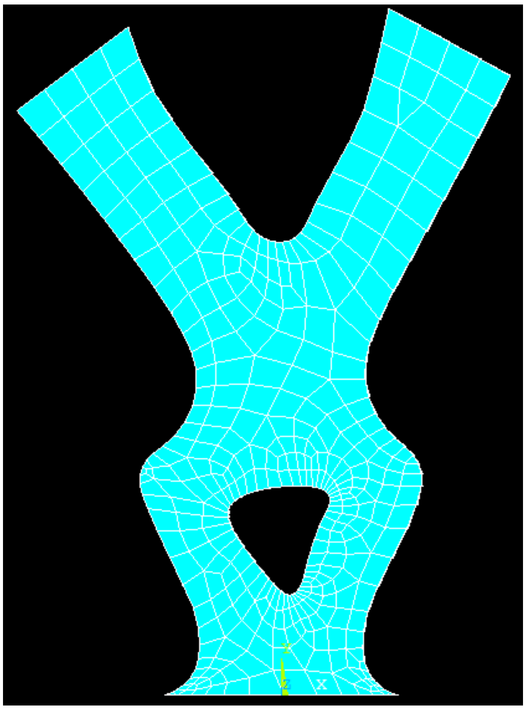


Figure 6.4. A SmartSized element mesh

a smaller element is needed and where the specified element size will suffice. As a result, in locations of the model where curves of surfaces become more complex, the element size is adjusted to cope with this. This is displayed in Figure 6.4 where a Special Node was created using *VisionNode*. It can clearly be seen that the mesh is denser near the hole in the node and near the curved edges. This is what SmartSizing does. Note that a meshed element is never larger than the earlier explained element size. A standard element size of 150 mm is used in *VisionNode*. However, this value can be adjusted by users of the program.

In *VisionNode*, the loads on a Special Node consist of an axial compressive force, a bending moment and a shear-force. These forces are transferred from the surrounding structure to the Special Node through the contact surface between the two. In *VisionNode*, the axial compressive force is translated to a surface pressure load, by dividing the force through the area of the contact surface, and applied to the FEM. This is also true for the shear-force which is applied as a surface load to the contact surface through the SURF154 element. See Figure 6.5. The bending moment is applied to the Special Node by using a so called super-element and two concentrated point loads. The super-elements are basically solid blocks of steel, placed on top of the contact surface. By applying two point loads to the super-element, the bending moment i simulated. The magnitude of the point loads is calculated by dividing the bending moment by the distance between the two point loads. Using the super-element and the point-loads, the bending stresses are correctly transferred to the Special Node. See Figure 6.6. The boundary condition in the FEM is that the base of the Special Node, the contact surface between Special Node and foundation, is restricted to move or rotate in all degrees of freedom. See Figure 6.7.

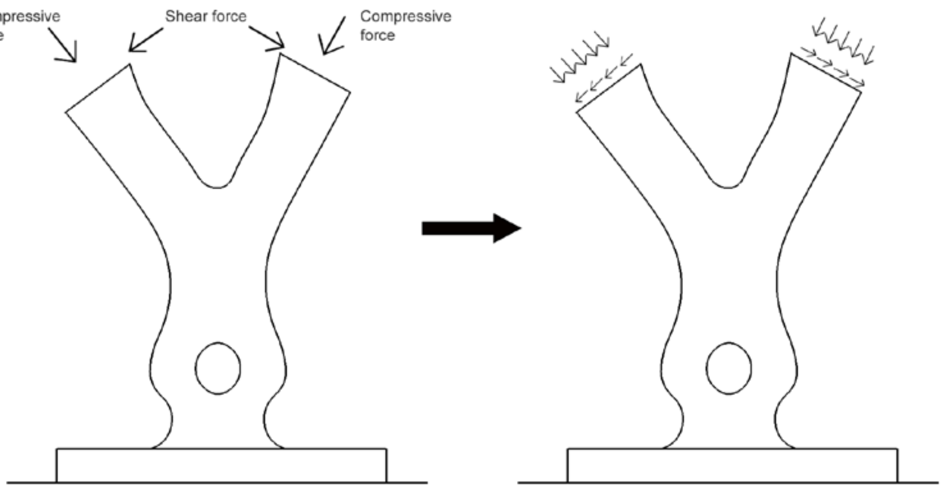


Figure 6.5. Forces to distributed pressure loads

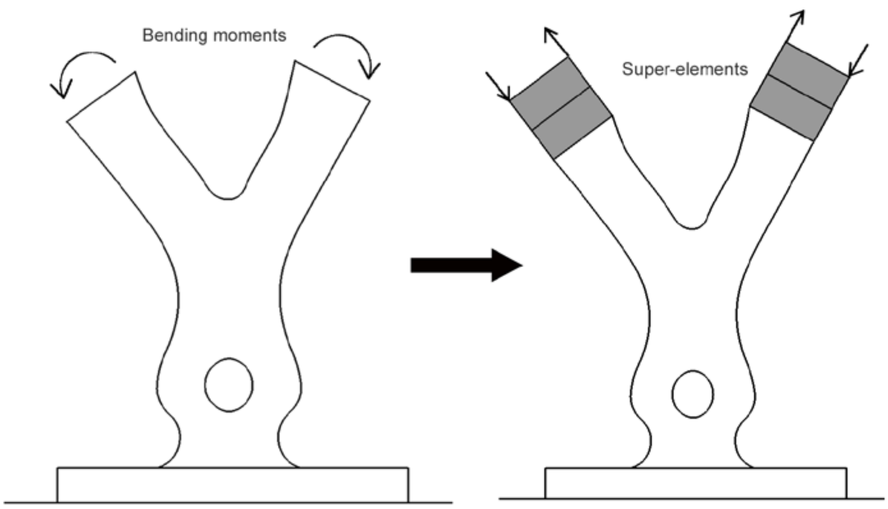


Figure 6.6. Bending moments to point-loads using Super-elements

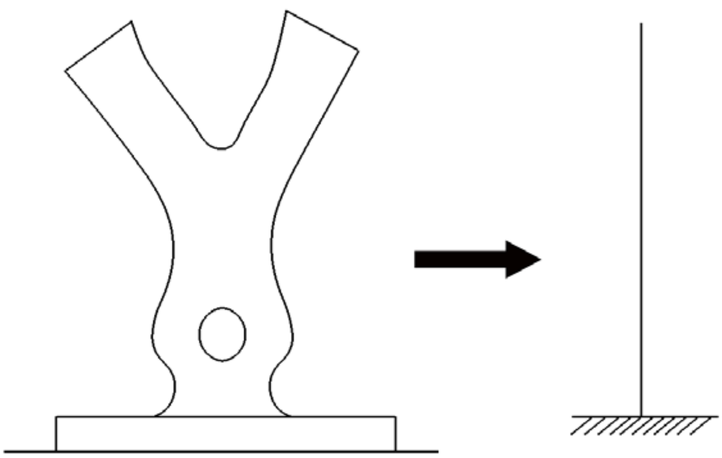


Figure 6.7. Boundary condition for Special Node

6.2. Implementation and operation in VisionNode

The FEM is created by *VisionNode* through the APDL-scripts, as was explained in Chapter 4. These scripts contain all the information that Ansys needs in order to create the FEM of a Special Node. After the *VisionNode* requested model has been created by Ansys, all properties are assigned to the model which have been explained in Section 6.1. When this is complete, Ansys starts the FEM solver and the structural calculations for the FEM are conducted. The amount of time that is required during the solving process, depends on the meshing element size and the type of the Special Node geometry that has been created by *VisionNode*'s geometric-modeller.

VisionNode has the ability to create two different kinds of Special Nodes in the FEM which are based on the same geometrical data. The first kind is a 2D-geometry of the Special Node which consists of an extruded area, thus creating a solid. The second kind is a 3D-geometry of the Special Node that does not involve extruding but involves so called lofting of surfaces.

Every Special Node is initially created in two dimensions in the form of NURBS-curves that are extracted from the design-variables. This NURBS representation of the Special Node geometry is exported to Ansys through the APDL-scripts. When *VisionNode* is creating a 2D-geometry of a Special Node this NURBS model is exported to Ansys which in turn creates a surface-area out of the NURBS-curves. See Figure 6.8. To create a 2D-geometry this area is extrude by the depth of the Special Node. As a result, a solid is created which is meshed and solved by Ansys. See Figure 6.9.

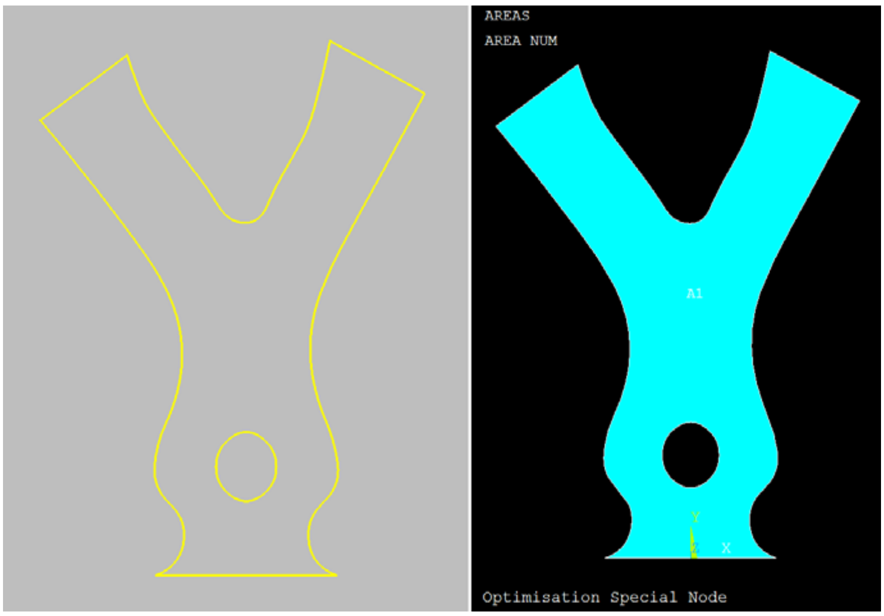


Figure 6.8. Area created out of the NURBS representation of the Special Node. Left: NURBS-model. Right: Area created by Ansys

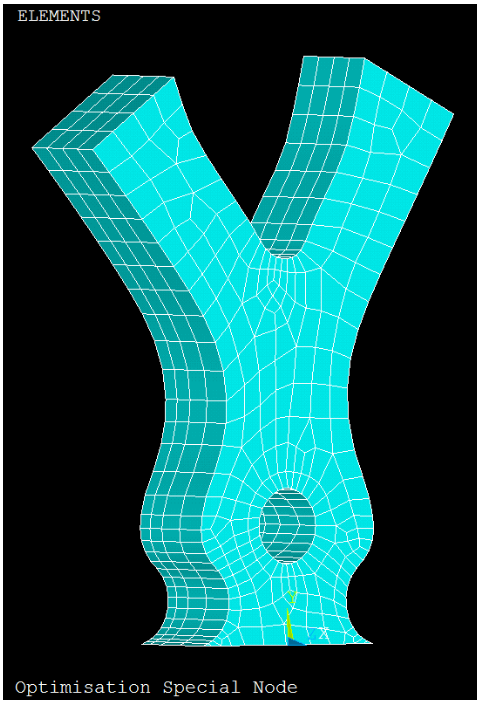


Figure 6.9. Solid created by extruding the area

VisionNode also has the ability to create a 3D-geometry of the Special Node. When generating this geometry, there is no NURBS representation of the Special Node exported to Anysy. Instead, the entire solid is created within Rhinoceros using its loft-surface functionality. This lofted surface model of the Special Node is then exported to Ansys which directly meshes the solid and executes the FEM solver. See Figure 6.10 for the representation of a loft-surface model of a Special Node and the FEM of that same Special Node.

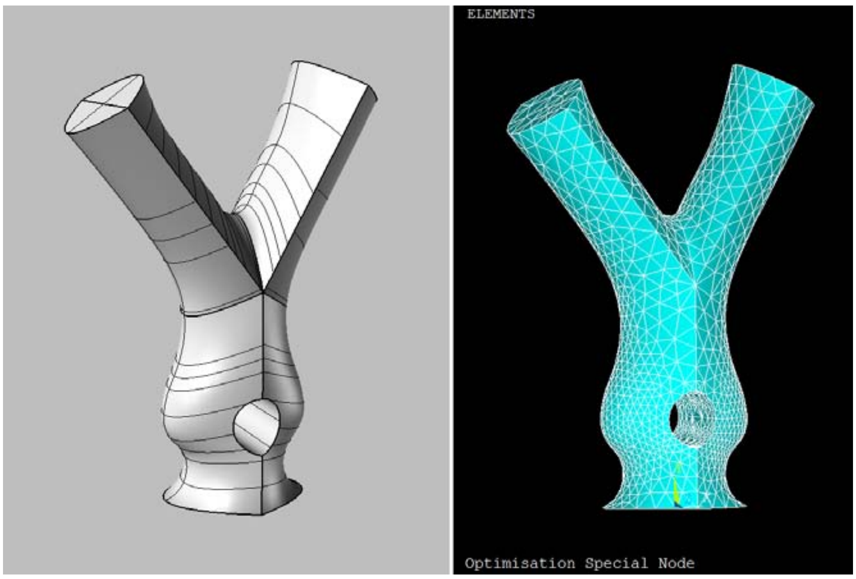


Figure 6.10. Lofted-surface model to meshed solid in FEM.
Left: lofted-surface. Right: FEM solid

The purpose for the switching between the 2D- and 3D-geometries, has to do with the time required during the solving-process of the FEM. The time required for the solving of a 2D-geometry is about 10-30 seconds while time required for the solving of a 3D-geometry is about 1-10 minutes. To speed up the optimisation process, only 3D-geometries are requested and created by *VisionNode* when the chance of obtaining a feasible optimisation solution is real. If during every iteration step, a 3D-geometry were to be created, the optimisation process would take up to an entire day instead of a couple of hours. For this reason, first 2D-geometries are calculated for the most part of the optimisation process and after that, when the solutions is nearly converged, the 3D-geometries are created to obtain the final designs of the Special Nodes. A solution is considered feasible when it satisfies the Mohr-Coulomb-Failure-Check. For the 2D-FEM, hexahedral meshing elements are used.

The Mohr-Coulomb-Failure-Check is based on the Mohr's Circle Theory and is also known as the internal friction theory. It is a structural failure-check that can be used for brittle materials such as concrete and UHPC, which are exposed to internal compressive and tensile stresses.
The equation of the failure check is:

$$\frac{\sigma_1}{\sigma_t} + \frac{\sigma_3}{\sigma_c} \leq 1$$

$\sigma_1 =$ largest principal tensile stress in structural element
$\sigma_3 =$ largest principal compressive stress in structural element
$\sigma_t =$ tensile capacity of material
$\sigma_c =$ compressive capacity of material

The value of the Mohr-Coulomb-Failure-Check is calculated by *VisionNode* after extracting the largest principal stress S1, which is the tensile stress, and the largest principal stress S3, which is the compressive stress, from the FEM. Through dividing these extracted values by the material capacities, the value of the Mohr-Coulomb-Failure-Check is obtained.

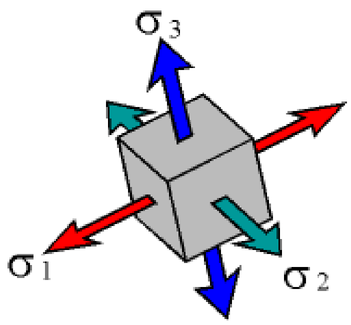


Figure 6.11. The principal stresses in a 3D stress scenario (Ansys Theory, 2007)

When a Special Node satisfies the Mohr-Coulomb-Failure-Check, it means that the Special Node does not yield under the forces to which it is exposed. Such a Special Node is considered as a feasible solution by the optimisation algorithm. The feasible solution with the smallest volume, is presented as the optimum solution.

Both the magnitude of the volume is calculated by Ansys and together with the magnitude of the Mohr-Coulomb-Failure-Check, these values are stored in a text-file under the name of a DAKOTA results-file. DAKOTA uses this data during the optimisation process and creates a new parameters-file for the geometric-modeller, based on the FEM results passed over by Ansys.

6.3. Conclusion

In this chapter, the properties and characteristics of the FEM used by *VisionNode* have been presented and explained. A fully functioning FEM has been successfully implemented in *VisionNode*.

The finite-element types, the meshing-element types and the meshing-element size that have been used for the FEM, have been found to function correctly for the Special Nodes. All material and geometrical properties are correctly extracted from *VisionNode* and uploaded into the FEM. Both the 2D-geometry and the 3D-geometry of the Special Node are being created and solved correctly by the FEM.

The Mohr-Coulomb-Failure-Check is calculated by *VisionNode* and the values are being calculated correctly.

Chapter 7.

Results & Evaluation

VisionNode is the product of this research project and its purpose is to determine whether it is technically feasible to create Special Nodes, as defined in this project, using UHPC with fibre reinforcement and without passive reinforcement as a construction material. In essence, *VisionNode* serves as the means to a purpose. Using *VisionNode*, several Special Nodes in UHPC have been created and the results are presented in the first part of this chapter. In addition, the operations and the methods of *VisionNode* have been tested and evaluated. In the second part of this chapter, the evaluation results are presented and discussed. The chapter is finalised with conclusions.

7.1. Results

In the first and second chapter of this report, it has been explained that an existing building structure would serve as an example structure for this research project. To demonstrate *VisionNode* for a possible real life application, the Yas-Hotel building in Abu Dhabi was chosen to serve as this example structure for which a Special Node was to be created. *VisionNode* has been successfully utilized for this purpose and several optimised designs of a Special Node have been created for the Yas-Hotel. The obtained results are presented in this section.

Using *VisionNode's* interactive graphical-user-interface, design preferences and modifications have been made prior to the optimisation process. These modifications consist of connection coordinates, connection dimensions and remove- and add-objects. The pre-optimised design including the design modifications is referred to as the 'preliminary design' in the continuation of this chapter. A total of two preliminary designs have been made which have resulted in two Special Nodes.

For every preliminary design, multiple optimisations have been run in order to have multiple optimum solutions available to those from. Another reason to run multiple optimisations, was to establish a working standard set of parameters for the genetic algorithm (GA) which produced acceptable results for any preliminary design. This standard parameter-set serves as a solid starting point for any optimisation. Where needed, small modifications have been made to the standard GA parameter-set in order to make the optimisation of each preliminary design, run more efficiently. Information on the GA parameter-set can be found in Chapter 5.

After numerous test optimisations, the standard GA parameter-set was determined and is presented in Table 7.1.

Seed	Random Five Digit Integer
Initial Population Size	20
Initialisation Type	Unique Random
Crossover Type	Shuffle Random
Crossover Rate	0,9
Mutation Type	Replace Uniform
Mutation Rate	0,1
Fitness Type	Merit Function
Replacement Type	Favor Feasible
Convergence Type	Best Fitness Tracker
Percentage Change	0,01
Number of Parents	4
Number of Offspring	4
Maximum Iterations	2500
Maximum Number of Generations	100

Table 7.1. The standard genetic algorithm parameter-set

7.1.1. Preliminary Design 1

The first preliminary design was created using one Remove-Object and is presented in Figure 7.1. The corresponding *VisionNode* input parameters are displayed in Figure 7.2. On the left of Figure 7.1 the minimum-geometry of the preliminary design is shown. On the right, the minimum- and the maximum-geometry are displayed. The red line is the lower-geometrical-boundary for the optimisation algorithm and the yellow line depicts the upper-geometrical-boundary. For more information on the minimum- and maximum-geometry, the reader is referred to Chapter 4.

With regard to the *VisionNode* input parameters from Figure 7.2, note the following:

- UHPC mixture Ductal C170/200
- Young's Modulus = 60.000 MPa
- Width of the base (foot) of the node = 1000 mm
- Width of left connection = 600 mm
- Width of the Right connection = 600 m
- Depth of the node = 600 mm
- One remove-objects, no add-object
- Maximum distance = 500 mm
- Minimum distance = 200 mm
- Meshing element size = 150 mm
- Total height = 2956 mm
- Total width = 2129 mm

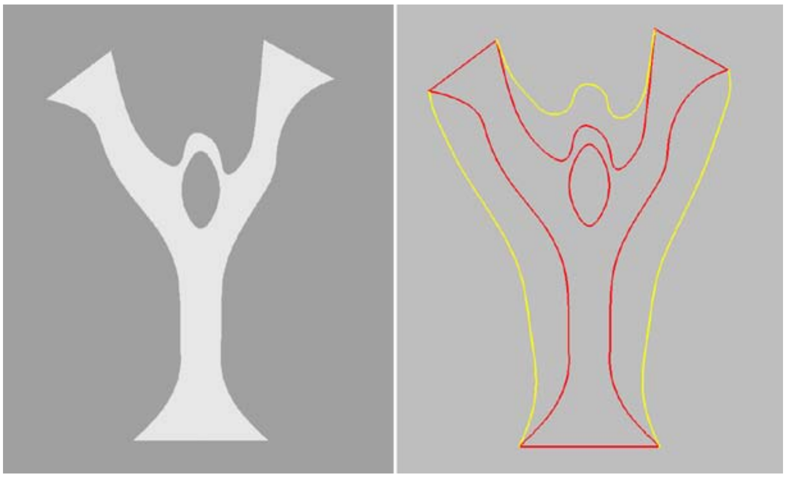


Figure 7.1. The minimum- and maximum geometry of the first preliminary design

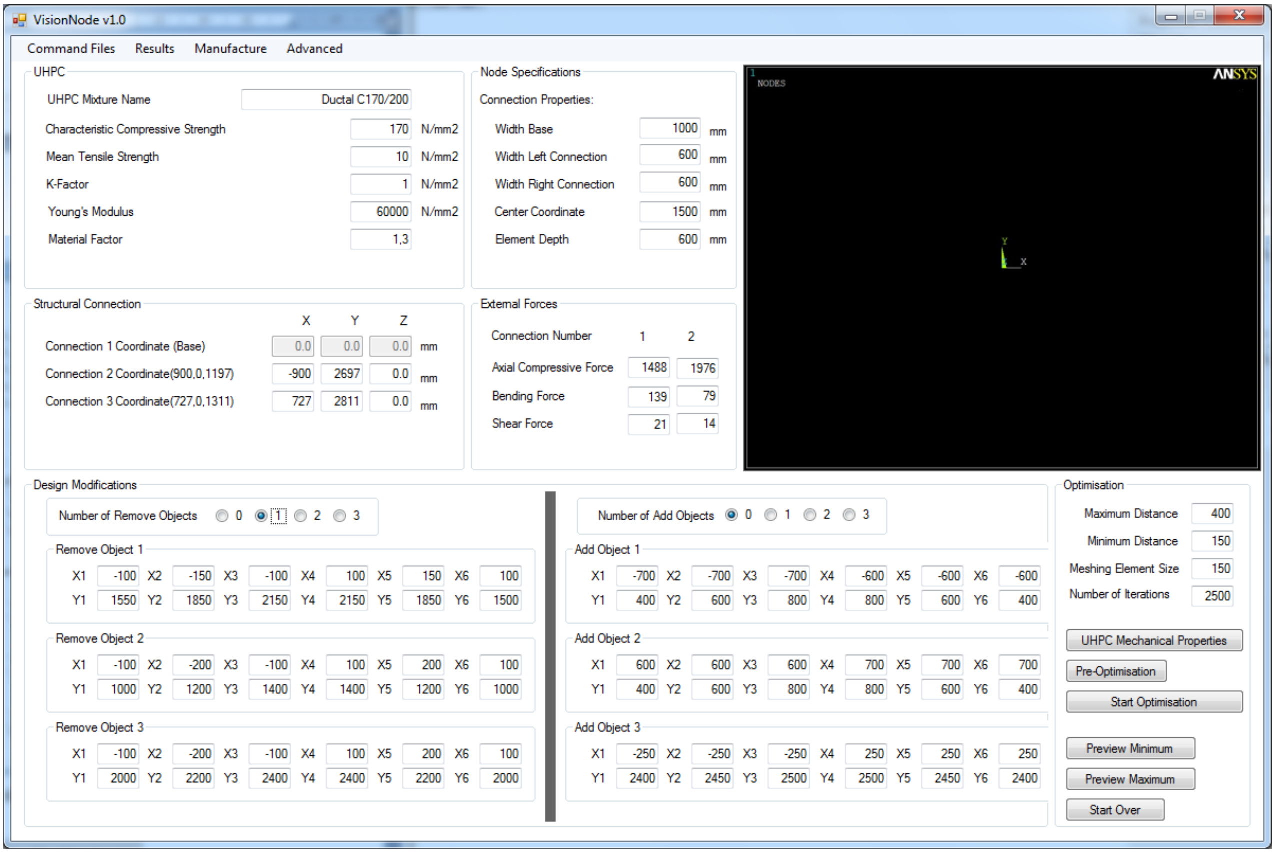


Figure 7.2. The VisionNode input-parameters for the first preliminary design

Seed	Random Five Digit Integer
Initial Population Size	23
Initialisation Type	Unique Random
Crossover Type	Shuffle Random
Crossover Rate	0,9
Mutation Type	Replace Uniform
Mutation Rate	0,11
Fitness Type	Merit Function
Replacement Type	Favor Feasible
Convergence Type	Best Fitness Tracker
Percentage Change	0,01
Number of Parents	4
Number of Offspring	4
Maximum Iterations	2500
Maximum Number of Generations	100

Table 7.2. The slightly modified genetic algorithm parameter-set for the first preliminary design

With the input parameters from Figure 7.2, several optimisations have been run using the standard GA parameter-set as a starting point. After several optimisations the genetic algorithm parameter-set that presented the best results was established and is shown in Table 7.2. It was found that increasing the mutation rate to 0,11 and the initial population size to 23 worked best. By increasing the mutation rate, the algorithm was allowed to randomize its search a little more than with the original mutation rate of 0,10. By increasing the population size, the algorithm was allowed to run more iterations for each generation and consequently increasing the amount of data before moving to the next generation. For this particular design, the algorithm seemed to need the extra randomisation and additional data. As a result, convergence of the solutions was more efficient. The optimisation process, however, took considerably longer.

The results of the optimisation are displayed in the diagrams in Figures 7.4 and 7.5. Figure 7.4 shows the boundary constraint of the Mohr-Coulomb-Failure-Check and Figure 7.5 shows the volume of the Special Node. Looking at the diagrams, it can clearly be seen that the algorithm puts the highest priority on satisfying the constraint and obtaining feasible solutions. It can also be seen that in order to reach the feasible solutions, the algorithm has to increase the volume of the node. This is a logical effect because the high stresses are caused by insufficient material. To lower the stresses, the algorithm increases the volume of the node. After 2063 iterations which took about three hours, the solutions converged and the optimisation process was completed.

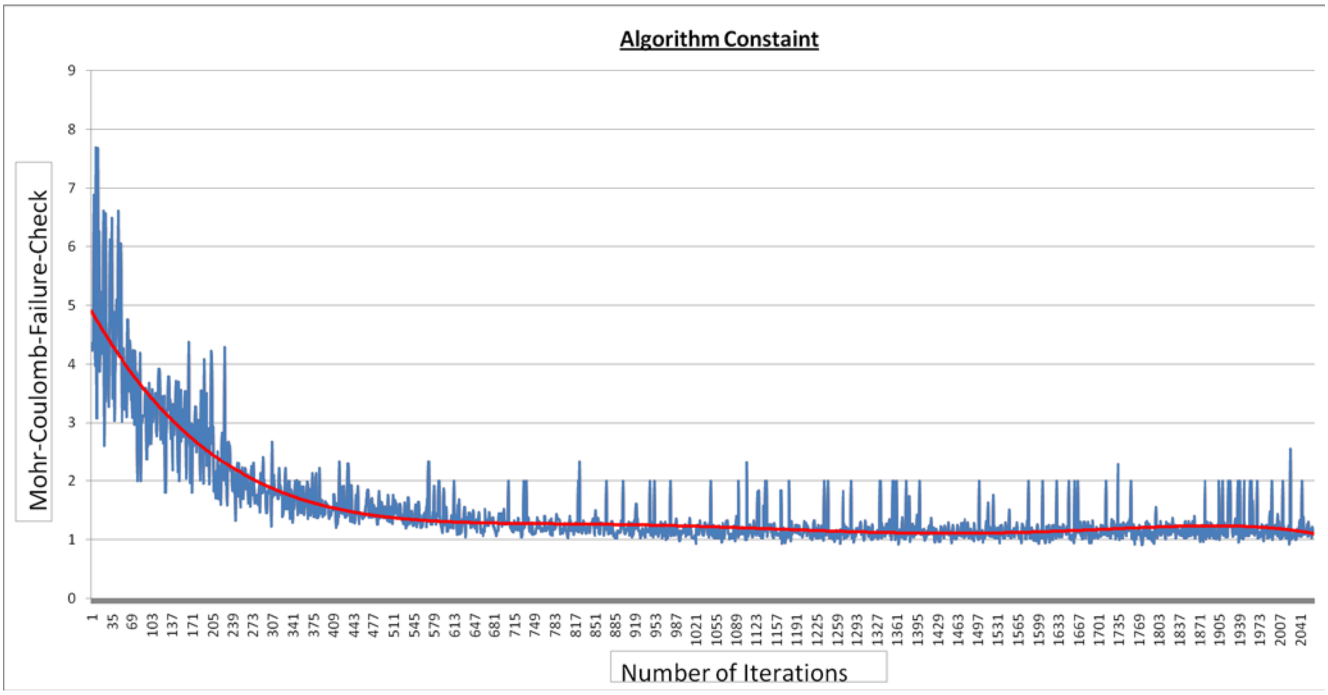


Figure 7.4. Optimisation constraint preliminary design 1

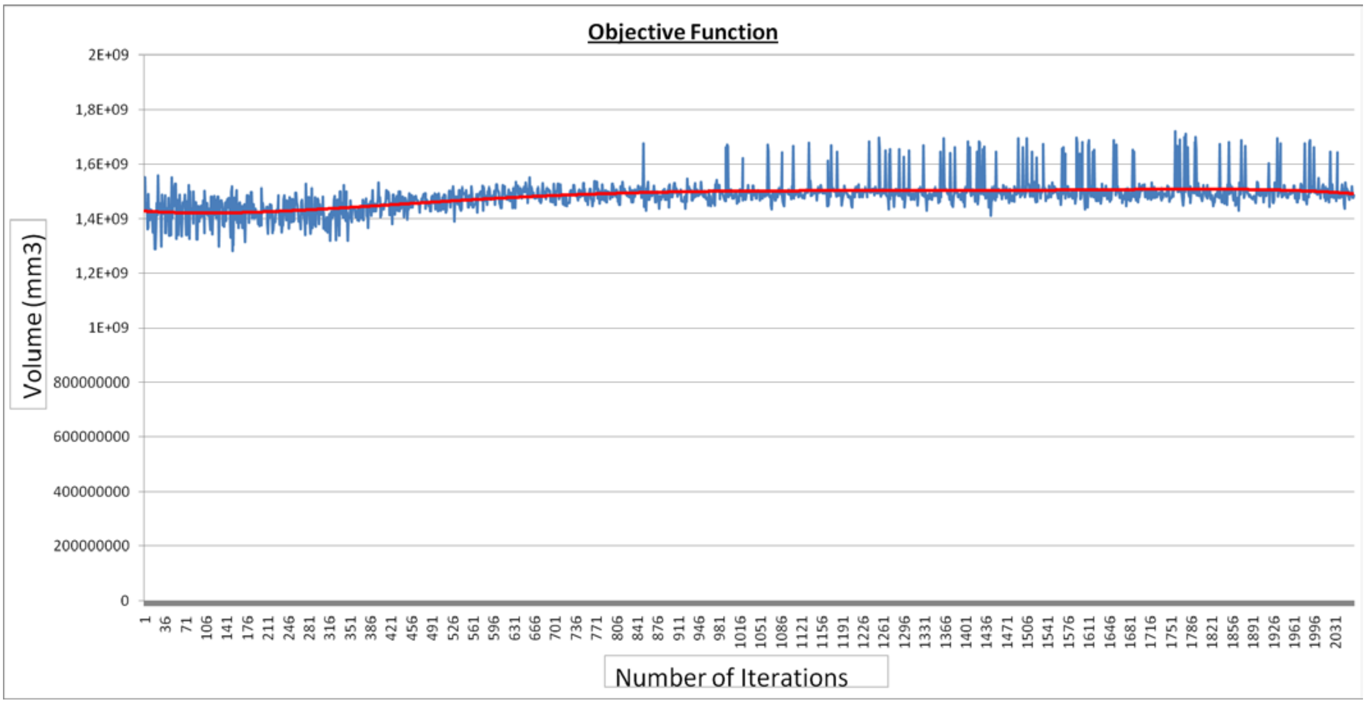


Figure 7.5. Optimisation objective function preliminary design 1

From all the feasible solutions, five with the lowest volumes have been extracted from the optimisation data and are presented in Table 7.3. The Mohr-Coulomb-Failure-Check is the constraint condition during the optimisation and the volume of the Special Node is the objective function which the optimisation is trying to minimize.

Mohr-Coulomb-Failure-Check	Volume (mm3)
0,934	1614436170
0,936	1623863310
0,977	1637171590
0,941	1638325000
0,980	1642839430

Table 7.3. Five best solutions from optimisation data

The solution with the smallest volume was taken form Table 7.3 and used to generate the final optimised preliminary design. This Special Node is displayed in Figure 7.6. The corresponding FEM calculation results are presented in Figure 7.7. On the left of this figure, the tensile stress distribution (S1) is shown and on the right the compressive stress distribution (S3).

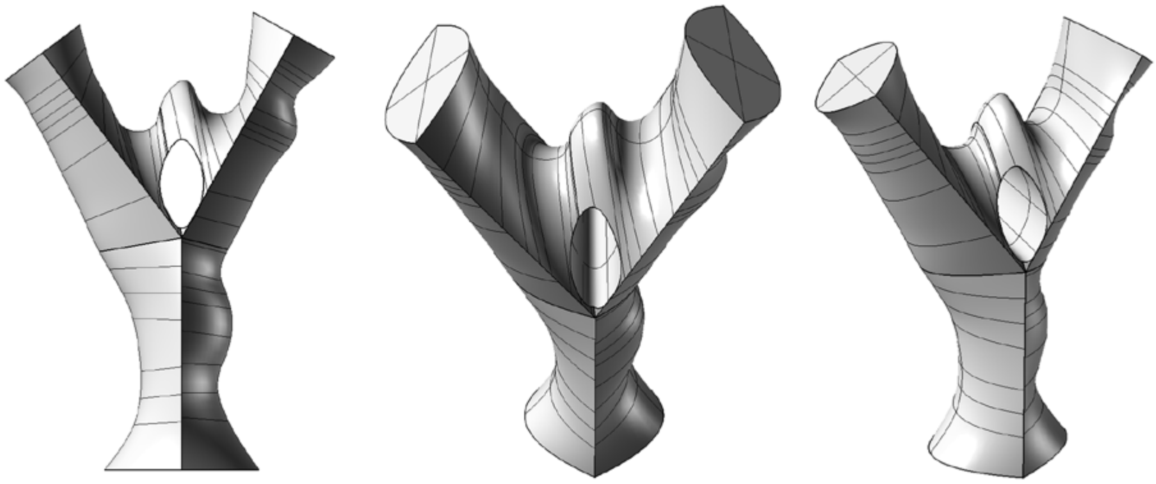


Figure 7.6. Special Node created by VisionNode out of preliminary design 1

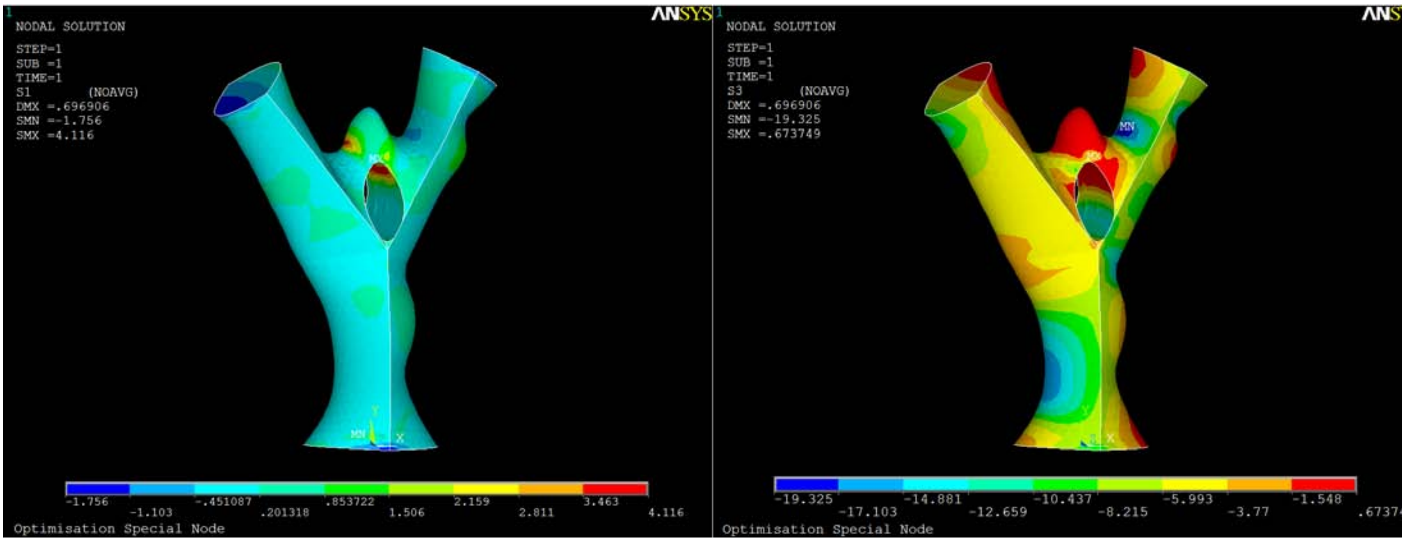


Figure 7.7. FEM for the optimum solution. Left: The tensile stress distribution in MPa. Right: The compressive stress distribution in MPa.

Volume of Special Node (mm^3)	1614436170
Maximum Tensile Stress (MPa)	4,116
Maximum Compressive Stress (MPa)	-19,325
Mohr-Coulomb-Failure-Check	0,9388
Optimisation Time	3 hours

Table 7.4. Optimum solution data for first preliminary design

7.1.2. Preliminary Design 2

The second preliminary design is presented in Figure 7.8 and the input parameters are presented in Figure 7.9. For this design two remove-objects have been used.

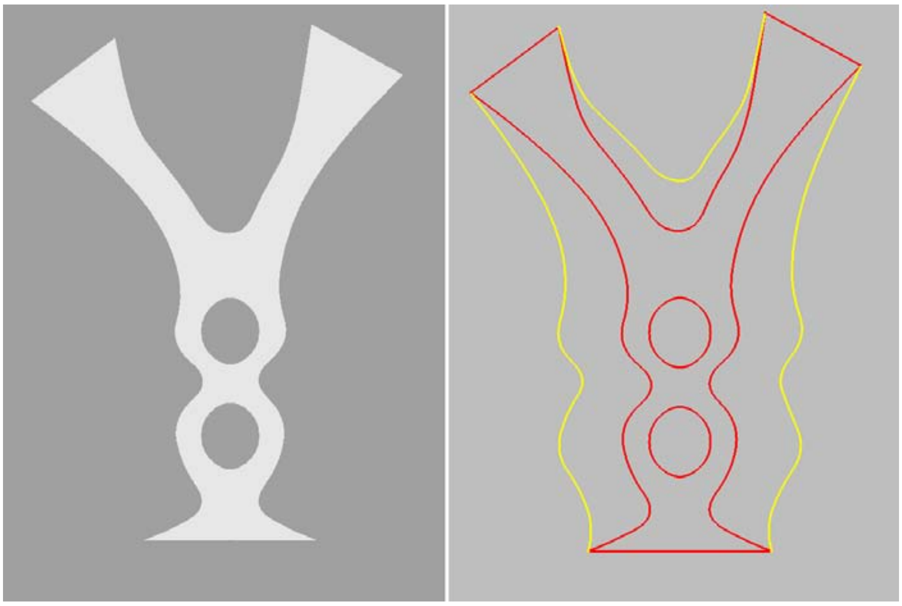


Figure 7.8. The minimum- and maximum geometry of the second preliminary design

Note the following input parameters from Figure 7.9 for this design:

- UHPC mixture Ductal C170/200
- Young's Modulus = 60.000 MPa
- Width of the base (foot) of the node = 1000 mm
- Width of left connection = 600 mm
- Width of the right connection = 600 m
- Depth of the node = 600 mm
- Two remove-objects, no add-object
- Maximum distance = 400 mm
- Minimum distance = 150 mm
- Meshing element size = 150 mm
- Total height = 2956 mm
- Total width = 2129 mm

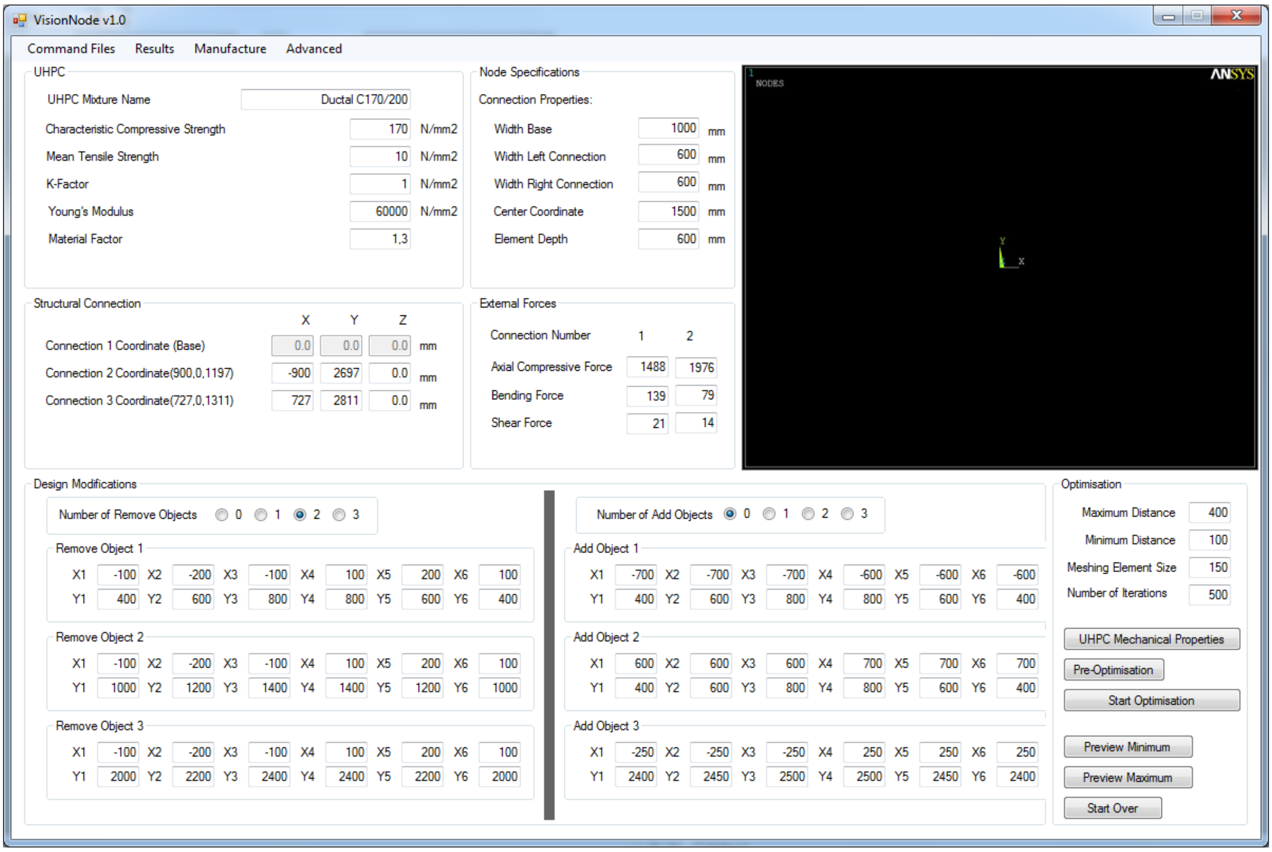


Figure 7.9. The VisionNode input-parameters for the second preliminary design

As was the case for the first preliminary design, a number of optimisations have been run taking the standard GA parameter-sets as a starting point.

After several optimisation runs with the input-parameters shown in Figure 7.9, it became clear that the optimisation algorithm was not converging enough to obtain a Mohr-Coulomb-Failure-Check smaller or equal to 1. The diagram in Figure 7.10 shows the algorithm constraint convergence during the optimisation. It can be seen that the solution is converging but never really gets near the desired value of 1. The reason for this non-convergence was found in the limited tensile capacity of the UHPC mixture C170/200. No matter what the algorithm tried, it could not reduce the tensile stress in the Special Node enough to satisfy the tensile capacity of this UHPC mixture. Adjusting the standard GA parameter-sets did not help either.

As the tensile stress was governing during this optimisation, a higher tensile strength for UHPC was chosen. From literature, a UHPC mixture was taken with a substantially higher

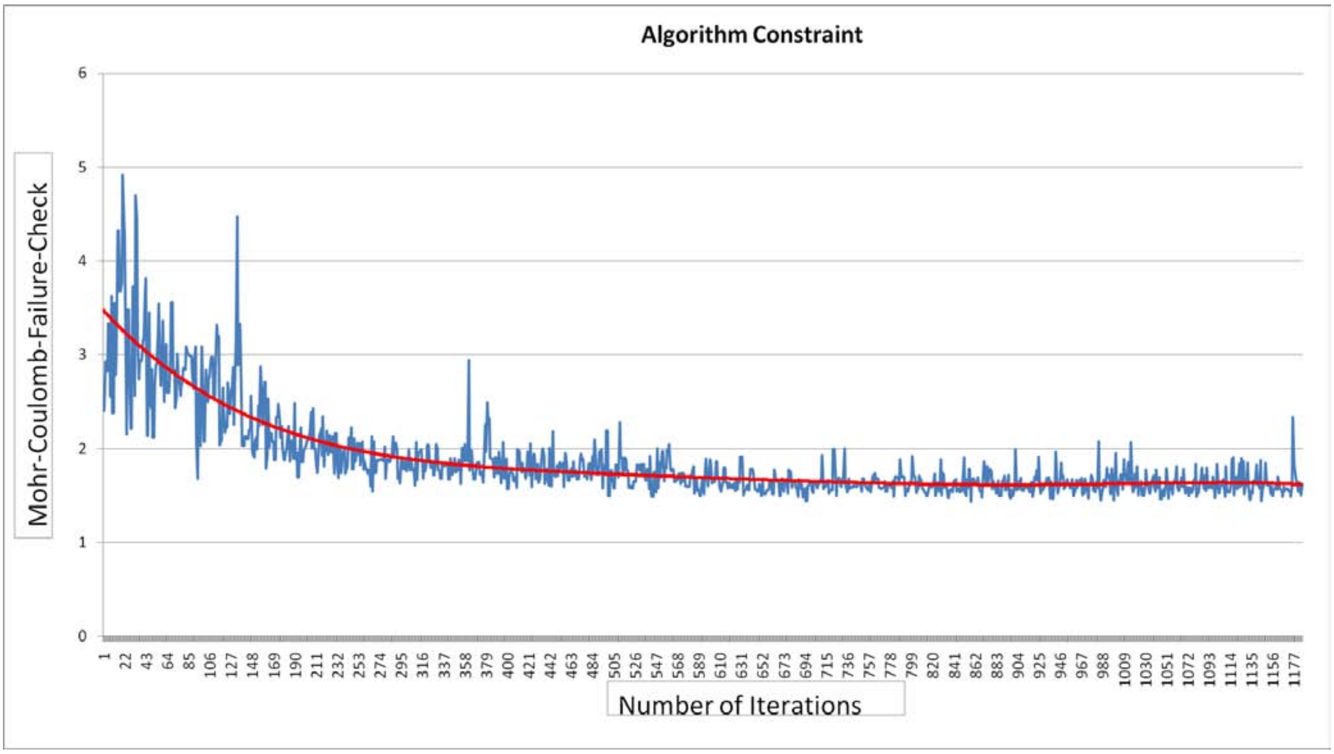


Figure 7.10. Non-converging algorithm constraint

tensile strength based on experiments conducted in 2008 by Wuest et al. During the experiments, several UHPC mixtures were tested and the tensile capacities were determined. The UHPC mixture which was designated UHPFRC-2, had fibre reinforcement content of 4-6 % and was found to have an average elastic tensile strength of 10-11 MPa (Wuest et al., 2008).

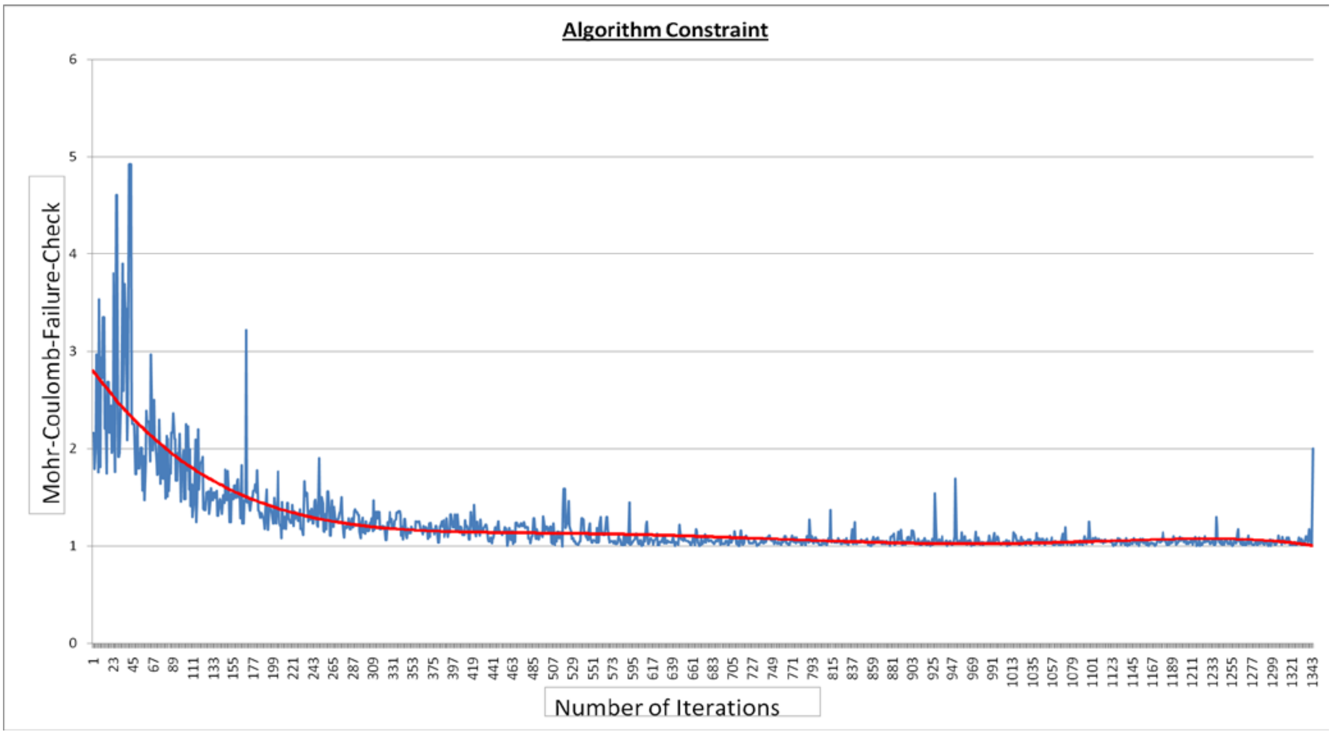


Figure 7.11. Optimisation Constraint preliminary design 2

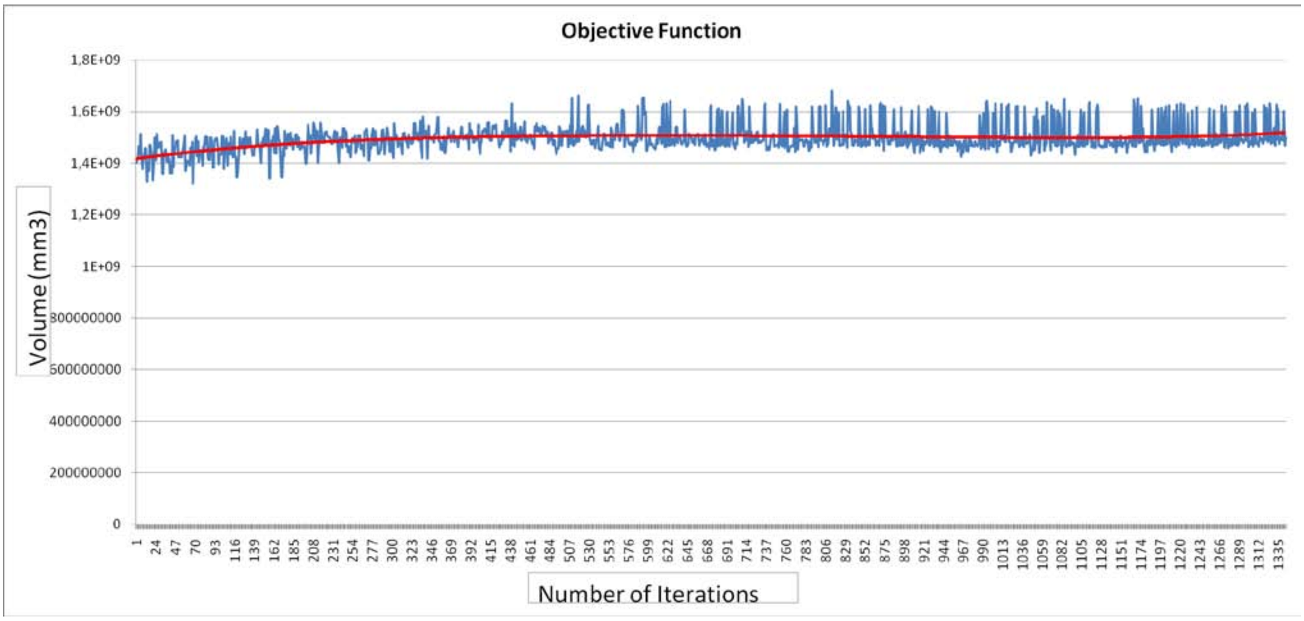


Figure 7.12. Objective Function preliminary design 2

Subsequently, the UHPFRC-2 mixture with its higher tensile capacity was input into *VisionNode* and new optimisations were performed on the second preliminary design. The results are presented in the diagrams in Figure 7.11 and Figure 7.12. The corresponding genetic algorithm parameter-set is displayed in Table 7.5. Note the lower mutation rate and population size.

Seed	Random Five Digit Integer
Initial Population Size	18
Initialisation Type	Unique Random
Crossover Type	Shuffle Random
Crossover Rate	0,9
Mutation Type	Replace Uniform
Mutation Rate	0,09
Fitness Type	Merit Function
Replacement Type	Favor Feasible
Convergence Type	Best Fitness Tracker
Percentage Change	0,01
Number of Parents	4
Number of Offspring	4
Maximum Iterations	2500
Maximum Number of Generations	100

Table 7.5. The slightly modified genetic algorithm parameter-set for the second preliminary design

As was the case during the first preliminary design, it can be clearly seen from Figure 7.11 and 7.12 that the algorithm immediately aims to satisfy the boundary constraint. As a result, the volume of the Special Node is increased slightly to lower the stress in the element. After 1328 iterations and 5 hours, the optimum solution was reached for this preliminary design.

Even though less iterations were required to reach the optimum in comparison to the first preliminary design, the optimisation took longer at 5 hours. The reason for this is that the optimisation needed a lot more 3D-geometry FEM calculations than in the first preliminary design. These calculations take up a lot more time than 2D-Special Node FEM calculations, which explains the higher optimisation time. This has been explained in Chapter 6.

From all the feasible solutions, the five best are displayed in Table 7.6. The solution with the smallest volume was taken from Table 7.6 and used to generate the final optimised preliminary design. The Mohr-Coulomb-Failure-Check for this particular solution is 1,00008204 which is slightly higher than the allowable ≤ 1. This difference is negligible and as a result, the solution is considered feasible.

The resulting Special Node is displayed in Figure 7.13 and the corresponding FEM calculation results are presented in Figure 7.14. On the left of this figure, the tensile stress distribution (S1) is shown and on the right the compressive stress distribution (S3).

Mohr-Coulomb-Failure-Check	Volume (mm3)
0,993729819	1662492250
1,00008204	1464257380
1,00030773	1467150310
1,00044701	1486288350
1,0005624	1491386710

Table 7.6. Five best solutions from optimisation data

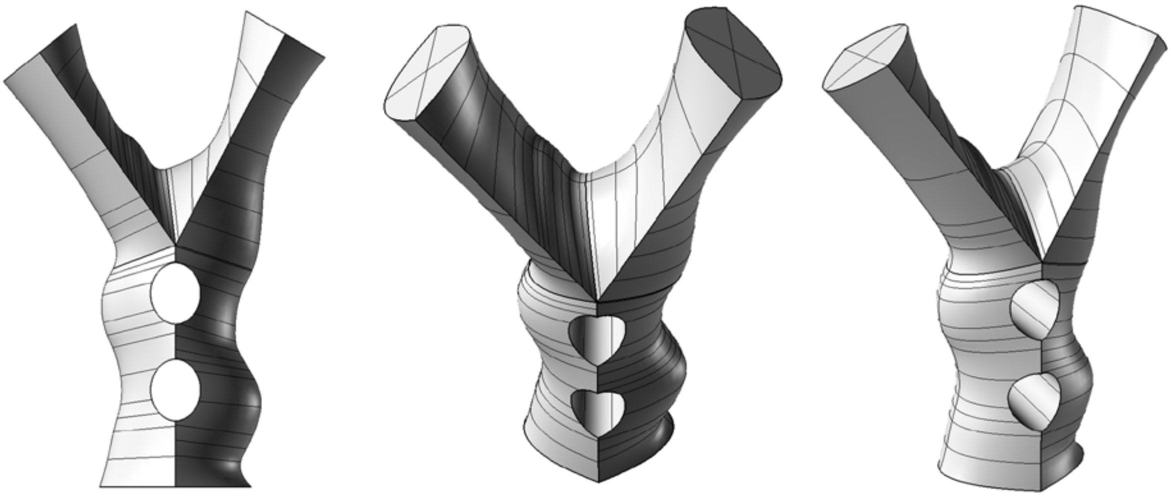


Figure 7.13. Special Node created by VisionNode out of preliminary design 2

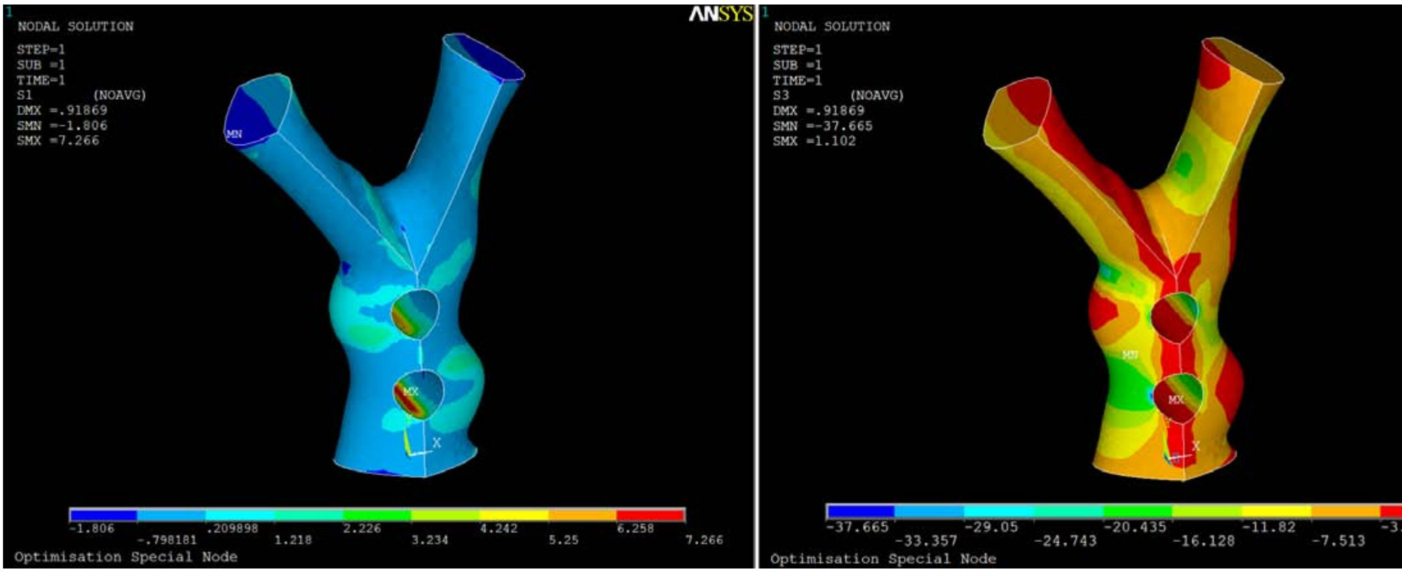


Figure 7.14. FEM for the optimum solution. Left: The tensile stress distribution in MPa. Right: The compressive stress distribution in MPa.

Volume of Special Node (mm^3)	1464257380
Maximum Tensile Stress (MPa)	7,266
Maximum Compressive Stress (MPa)	-37.67
Mohr-Coulomb-Failure-Check	1,000082
Optimisation Time	5 hours

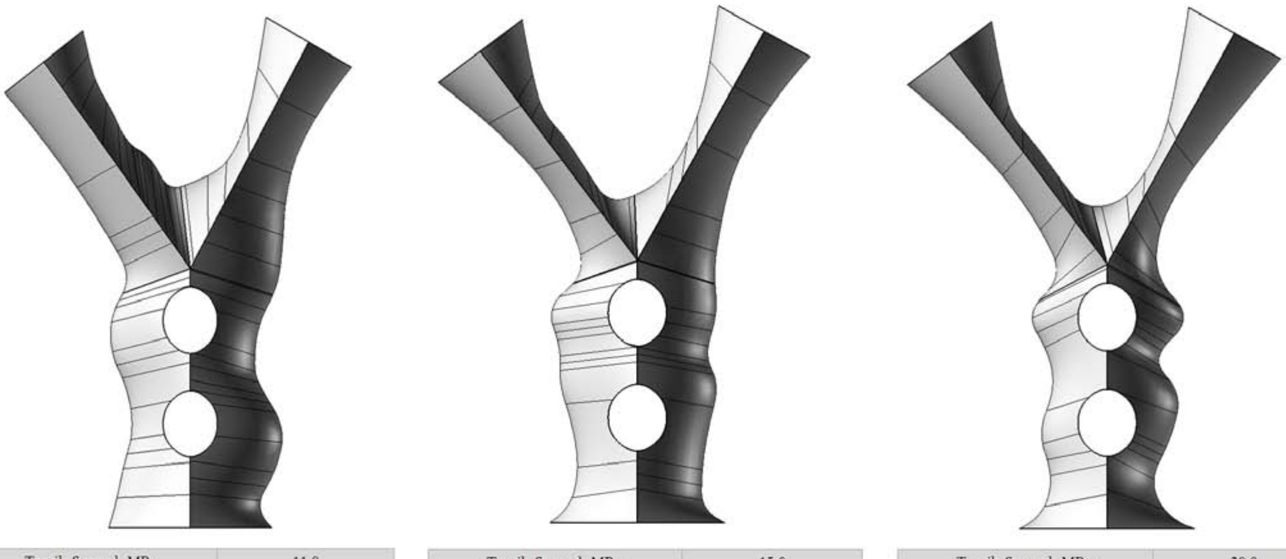
Table 7.7. Optimum solution data for first preliminary design

7.1.3. Effect of high tensile strength

In order to demonstrate what the effect of a high tensile is on the resulting Special Node, the second preliminary design was optimised an additional two times using a UHPC mixture with an increased tensile strength.

The second Special Node which is the result of preliminary design 2 from the previous sub-section, used an UHPC mixture with an maximum tensile strength of 10-11 MPa. In addition, the preliminary design 2 was optimised using a UHPC mixture with an elastic tensile strength of 15 MPa and 30 MPa. These values are imaginary and are chosen as such for this example only. There is no actual UHPC mixture in existence with an elastic tensile capacity of 30 MPa. The purpose of this example is to show what the optimisation algorithm does with a higher tensile strength.

The results are presented in Figure 7.15. From left to right, the elastic tensile strength is increased for each preliminary design. It can be seen that the Special Nodes with higher tensile strength have less volume and are more slender as a result.



Tensile Strength(MPa)	11,0
Volume of Special Node (mm³)	1464257380
Maximum Tensile Stress (MPa)	7,266
Maximum Compressive Stress (MPa)	-37.67
Mohr-Coulomb-Failure-Check	1,000082

Tensile Strength(MPa)	15,0
Volume of Special Node (mm³)	1368151684
Maximum Tensile Stress (MPa)	9,976
Maximum Compressive Stress (MPa)	-33.87
Mohr-Coulomb-Failure-Check	0,966

Tensile Strength(MPa)	30,0
Volume of Special Node (mm³)	1184695894
Maximum Tensile Stress (MPa)	16,6
Maximum Compressive Stress (MPa)	-45.64
Mohr-Coulomb-Failure-Check	0,961

Figure 7.15. Special Node with from same preliminary design with different tensile strengths

7.1.4. Two additional Special Nodes

When observing the Special Nodes created out of the first and second preliminary designs, it has been noticed that both resulting Special Nodes are non-symmetrical. The reason for this non-symmetry can be found in the fact that the angle of both arms and the loads transferred to both arms, are different. See Chapter 3 Section 3.4 for this. This was found to be the reason that the resulting Special Nodes are non-symmetrical.

To verify that there is no flaw in the software, two additional Special Nodes have been created using the input data from the first two preliminary designs as a starting point. Both preliminary designs were slightly altered to produce a node with the same angle in both arms and the same loads on both arms. After optimisation of these preliminary designs, the resulting Special Nodes were found to be nearly symmetrical. See Figure 7.15. It can be seen that for a symmetrical preliminary design with symmetrical loads on both arms, the created Special Nodes are symmetrical also. This example is meant to illustrate that the optimisation algorithm will create symmetrical Special Nodes when the input is also symmetrical.

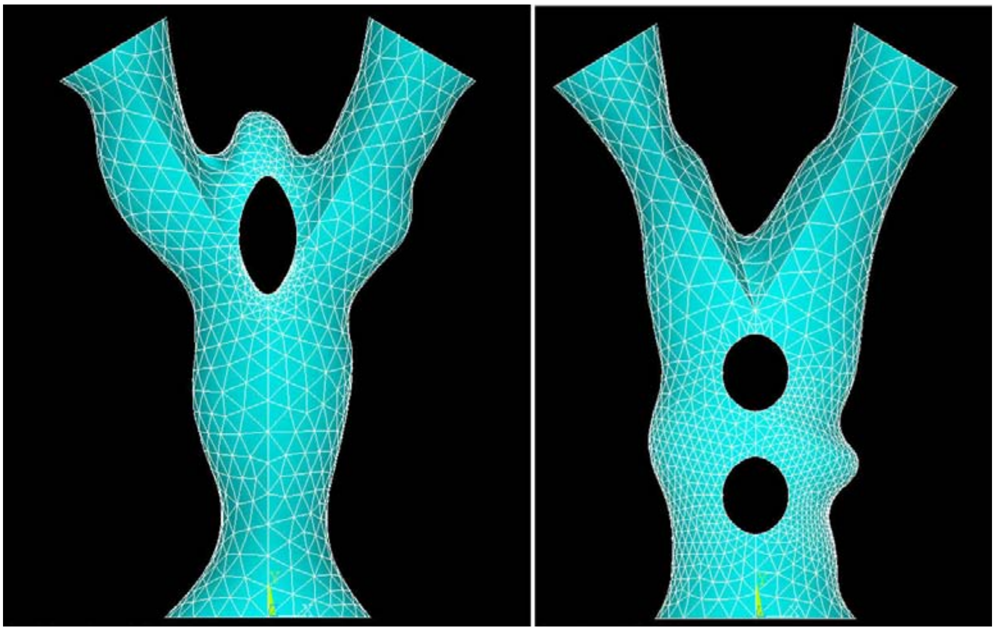


Figure 7.16. Special Node created from preliminary designs with symmetrical connection properties and external forces

7.2. Evaluation

In order to substantiate the results presented in the previous section, the operation of *VisionNode*, including the methods and the techniques it applies, must be tested and evaluated. If the tests are successful and the test-results comply with the expected results, than *VisionNode* can be considered fully operational. The evaluation of the following subjects is presented and discussed in this section:

- VisionNode software evaluation
- Finite-element analysis evaluation
- Optimisation algorithm evaluation

7.2.1. *VisionNode* Software Evaluation

In this sub-section the operation of *VisionNode* as computational software is tested and evaluated. The purpose of which is to establish whether the program is truly interactive and operates in accordance to the wishes of the user. Through a number of example cases, the tests have been conducted.

First of all, several preliminary designs without any remove- or add-objects are created with different connection coordinates and dimensions. Secondly, several preliminary designs with remove objects only have been created. Finally, several preliminary designs with add-objects are created. During all of these example cases, *VisionNode* has been checked whether it processes the input data correctly and created the user defined preliminary design.

VisionNode example case 1: Connection coordinates and dimensions

In the first example case, a total of three different preliminary are created using *VisionNode*. These are displayed in Figure 7.16. The corresponding input parameters are also presented in this figure.

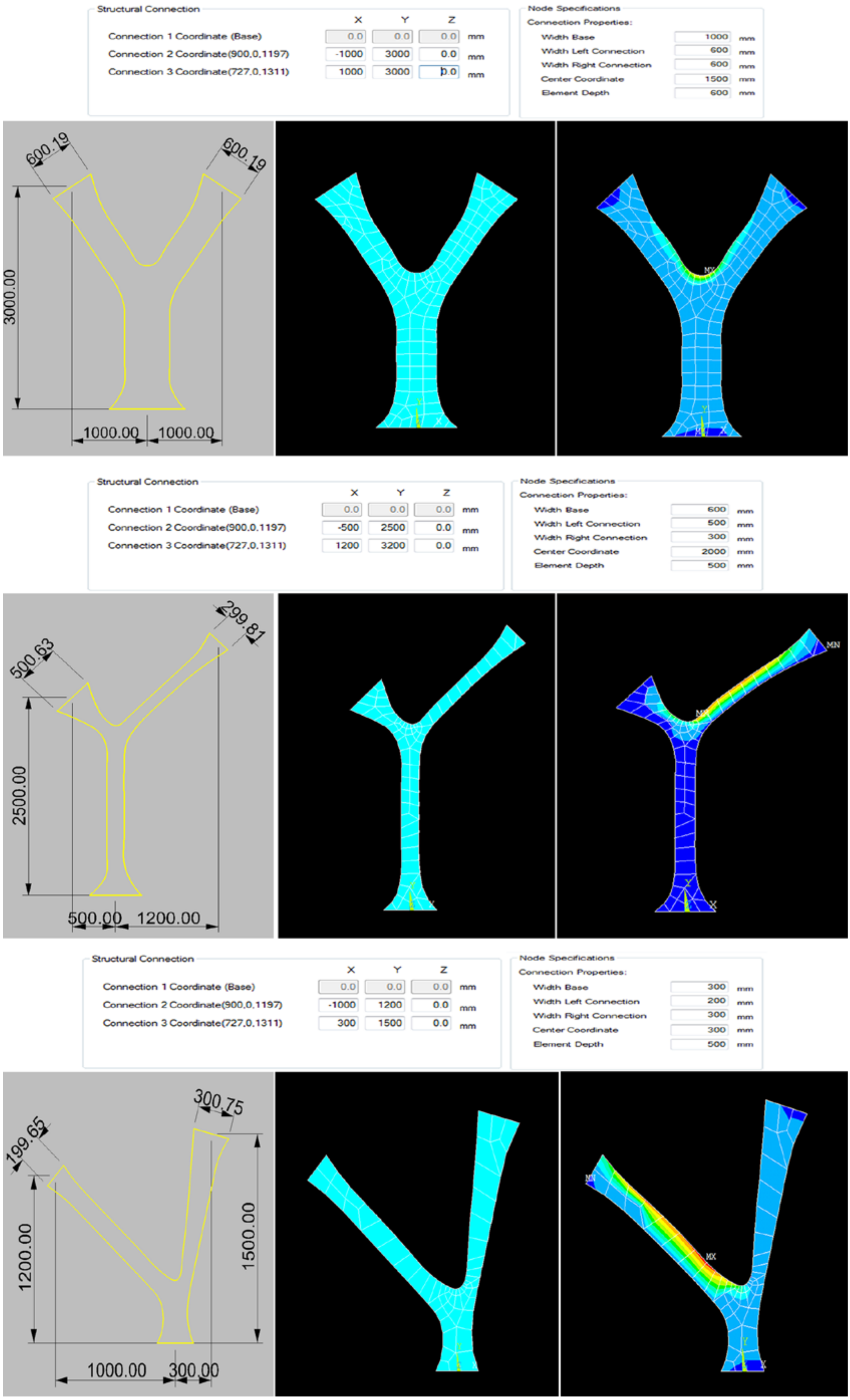


Figure 7.17. Three different designs with different input parameters

From Figure 7.16 it can be concluded that VisionNode processes the connectional input data correctly. The desired preliminary design was generated each time.

VisionNode example case 2: Special Node with remove-objects

In this example case, the capabilities of VisionNode in creating singular and multiple remove-objects is tested. During the test, the minimum and maximum shape-geometry of the preliminary designs are displayed with the corresponding Special Node.

In Figure 7.17 three different preliminary designs are shown with one, two and three remove–objects, respectively. For each preliminary design model, the minimum geometry as well as the maximum geometry shapes are presented. These shapes display the geometrical boundary conditions for the optimisation algorithm. This means, that the optimisation-generated Special Node cannot be smaller than the minimum shape and also cannot be larger than the maximum shape.

For the creation of the maximum geometry in the preliminary designs with remove-objects, *VisionNode* uses Distance-Factoring. This feature makes sure that the closer a remove-object is to an optimisation design-variable, the larger its maximum geometrical boundary becomes. Distance-Factoring is explained in detail in Chapter 4. The purpose and benefits of Distance-Factoring are also discussed in that chapter.

In Figure 7.17 the maximum geometry of the Special Nodes are shown with and without Distance-Factoring. From Figure 7.17 it becomes evident that Distance-Factoring is applied correctly in all cases. The maximum geometry without Distance-Factoring was obtained using an older version of *VisionNode.* Figure 7.18 show several other designs of Special Nodes with remove objects after 3D-geometry generation by Rhinoceros and Ansys.

From the previously shown figures it can be concluded that the software processes the location and geometry of remove-objects correctly. Equally important is that *VisionNode* modifies the surrounding geometry of the Special Nodes accordingly and correctly.

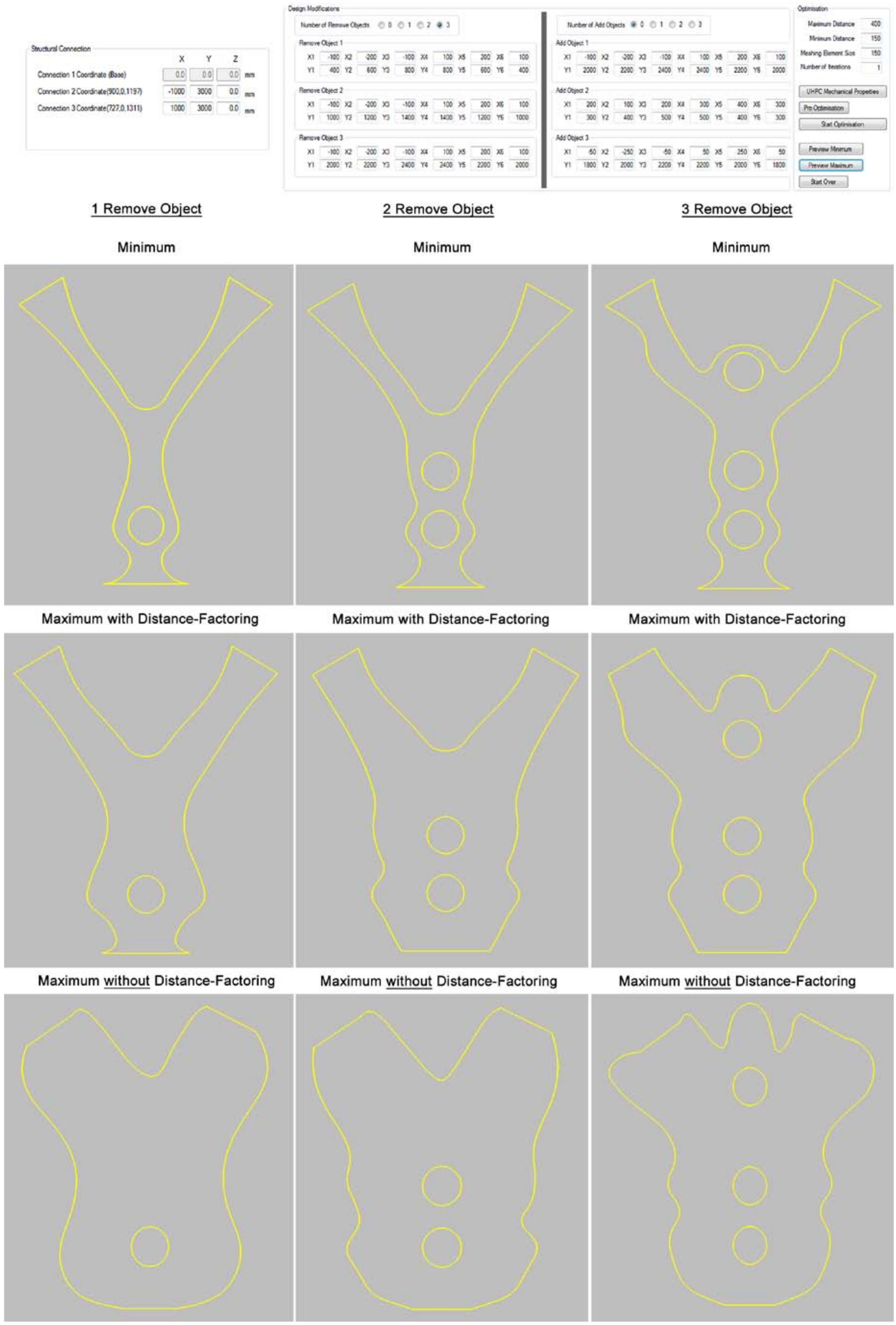


Figure 7.18. Three different designs with remove objects

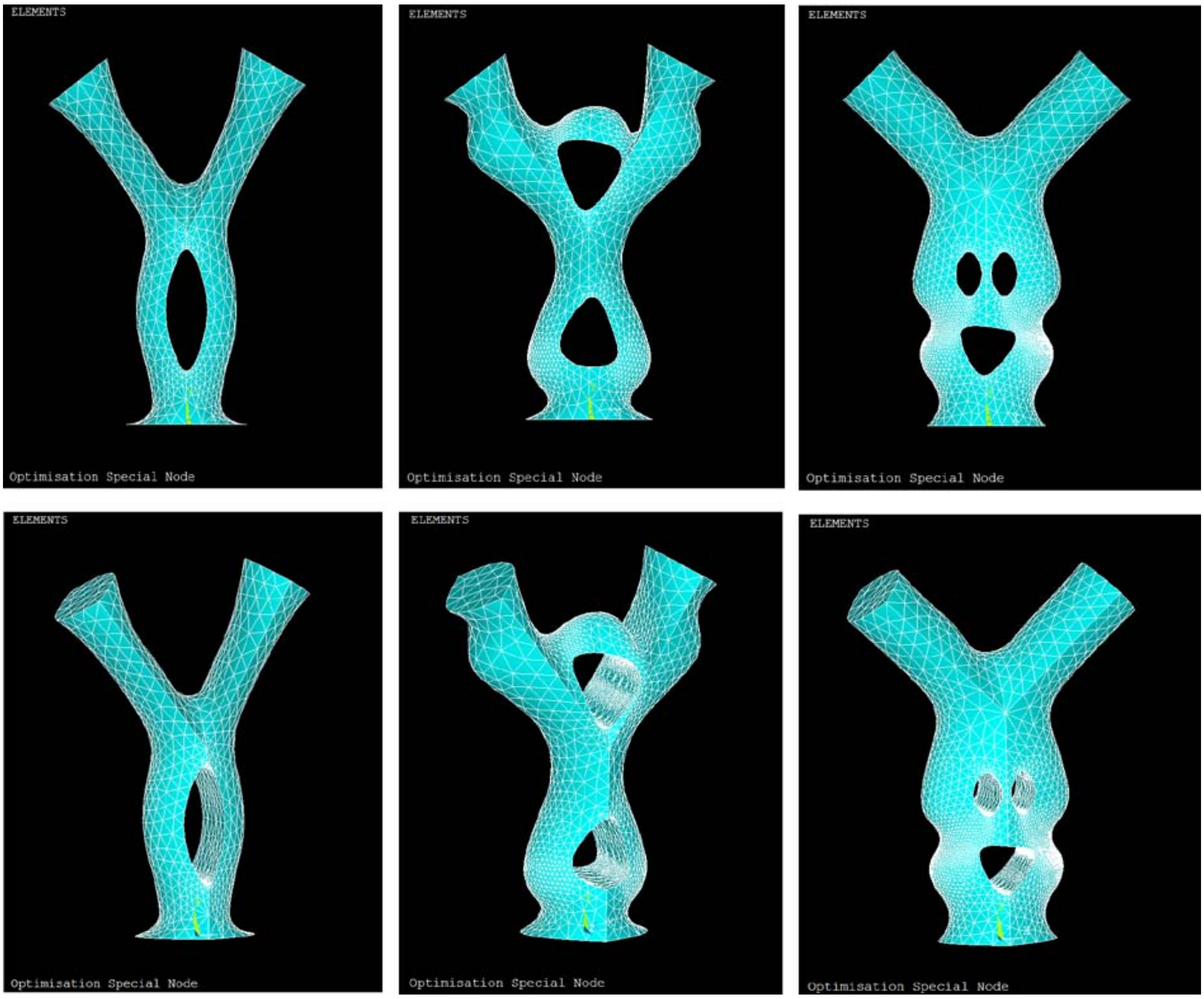


Figure 7.19. Additional designs with remove objects

VisionNode example case 3: Special Nodes with add-objects

In the third example case, the capabilities of *VisionNode* in creating remove- and add-objects is tested. The remove- and add-objects are applied simultaneously.

For easy comparison, the preliminary designs with two remove-objects from Figure 7.17 has been modified with add-objects. The modified Special Nodes is displayed in Figure 7.19. From top to bottom the following is displayed:

- The add-object dimensions
- The preliminary design prior to modification
- The location of the add-object represented as red rectangles
- The preliminary design including the add-objects

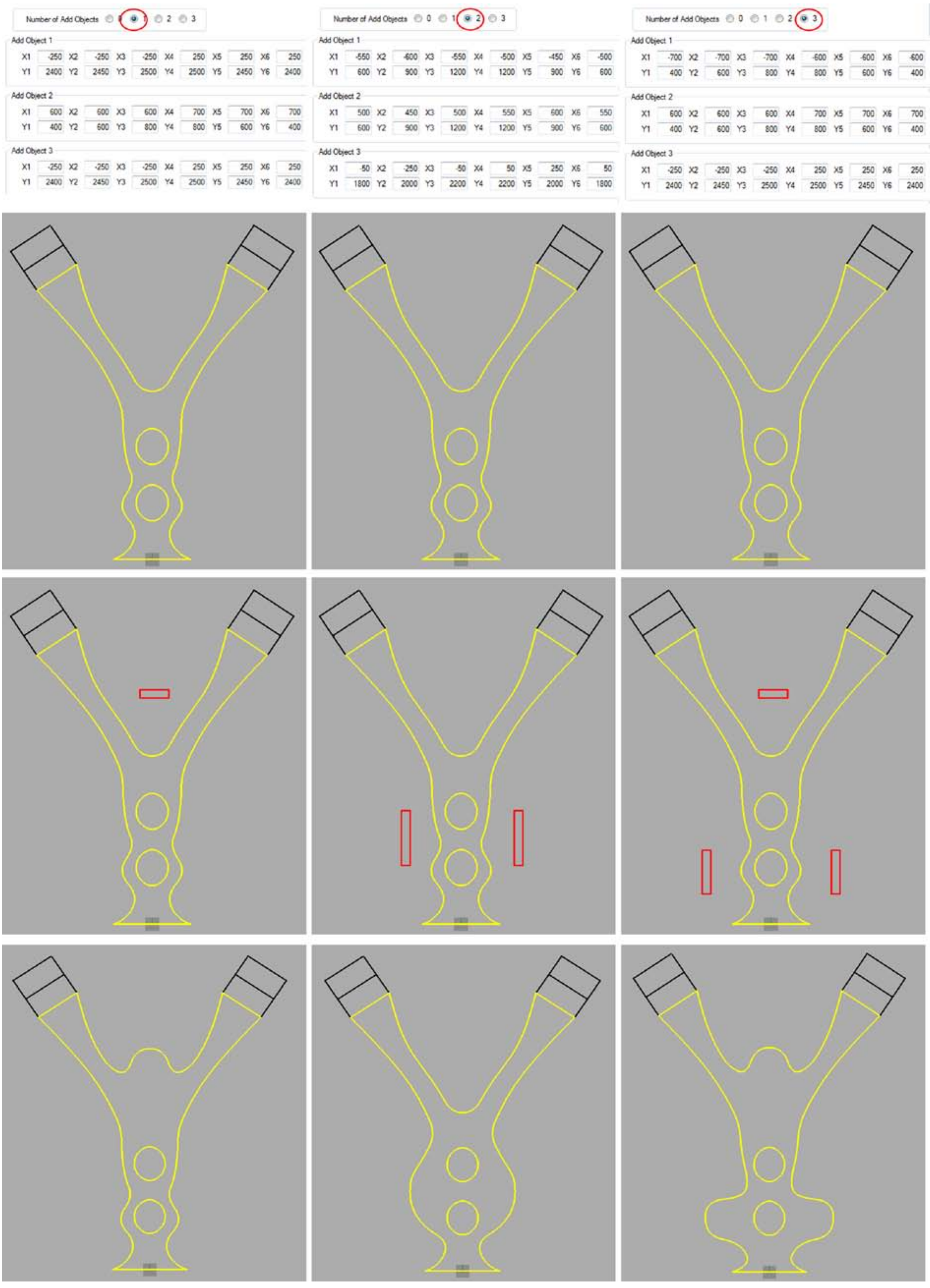


Figure 7.20. Preliminary designs with Add- and Remove-Objects

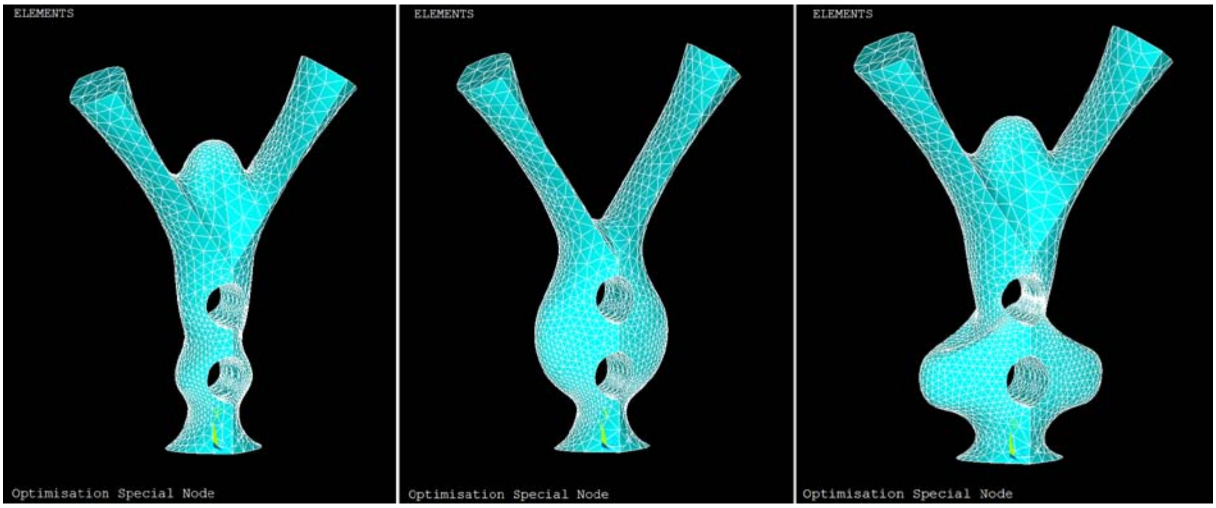


Figure 7.21. 3D-geometry of preliminary designs

In Figure 7.20 the same preliminary designs from Figure 7.19 are shown again but now after 3D-geometry generation by Rhinoceros and Ansys. It can be concluded that *VisionNode* places the Add-Objects on the right locations and modifies the geometry accordingly.

All the previous tests confirm that *VisionNode* as interactive software functions correctly and that the user has the ability to make design modifications and create a preliminary design, before the optimisation is run.

7.2.2. Finite-Element-Model evaluation

The successful operation of *VisionNode* and the optimisation algorithm it uses, relies heavily on the correctness of the Finite-Element-Model (FEM) that it utilised. For this reason, it is of the utmost importance that the FEM results are correct and accurate. In this sub-section the FEM used by *VisionNode* is evaluated though a number of example cases. The background and the properties of the FEM are discussed in Chapter 5. The example models used in this sub-section are created using this FEM.

The examples cases are simplistic in form and are not as complex as the Special Node calculation performed by Ansys during optimisation. However, if there is something wrong with the FEM, any errors or inaccuracies will immediately become evident, even if the example problem is simple in form.

A total of three FEM example cases are created and tested. First hand calculations are made, which are then compared to the FEM calculated values. A value difference of < 5% is considered to be within acceptable limits.

FEM example case 1: Reaction forces verification

The most elementary of all structural evaluations is the checking of the reaction forces at the support of the structural element. For this test a specific preliminary design model was created by *VisionNode*. The input dimensions and other properties are presented in Figure 7.21. Note the following properties for the preliminary design:

- UHPC Mixture C170/200
- Young's Modulus = 60000 N/mm^2
- The connection coordinates of both the node arms are identical which produces a symmetrical node.
- Element depth = 500 mm
- The minimum distance from the centre axis is 250 mm
- The maximum distance from the minimum coordinate is 0 mm
- Force distribution in left arm:
    - Compression force N_1 = 800 kN
    - Bending moment M_1 = 150 kNm
    - Shear force V_1 = 10 kN
- Force distribution in the right arm
    - Compression force N_2 = 1200 kN
    - Bending moment M_2 = 80 kNm
    - Shear force V_2 = 15 kN
- No gravity force within the node
- Meshing element size = 100 mm
- No Remove-Objects
- No Add-Objects

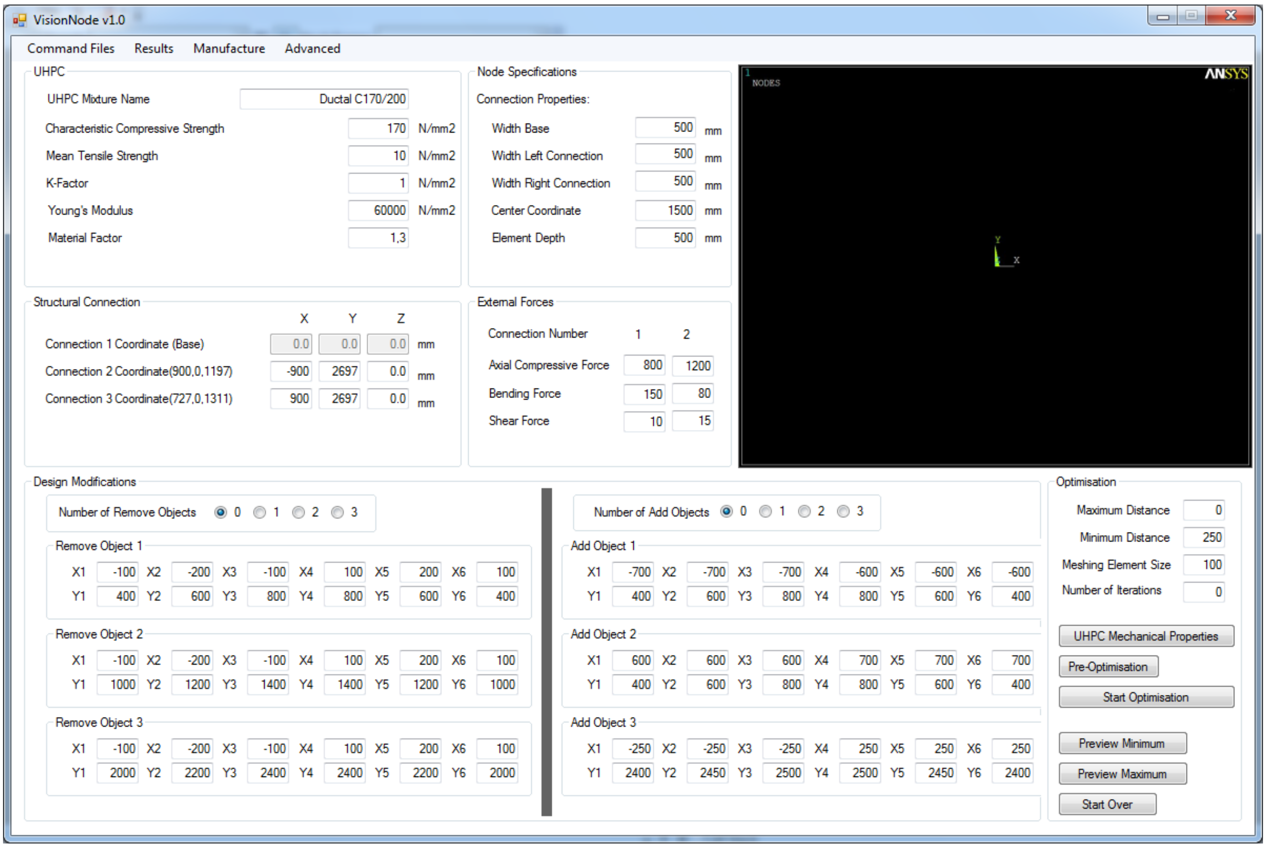


Figure 7.22. Input parameters for FEM example case 1

The tetrahedral element-type is chosen for the mesh in this example. See Chapter 6 for more information on this element type. As was explained in that chapter, the tetrahedral element-type gives slightly more accurate FEM results while increasing the calculation time with 400-800 %. Since the importance in this example case is accuracy, the tetrahedral element-type is used instead of the hexahedral element-type. By using the Preview-Minimum functionality, the specified node will be generated in Rhinoceros and loaded in Ansys afterwards.

The reaction forces in the X- and Y-directions at support are calculated by hand and later checked against the reaction forces calculated by the FEM. Note that the node is restrained in all degrees-of-freedom at support and that there are no forces in the Z-direction.

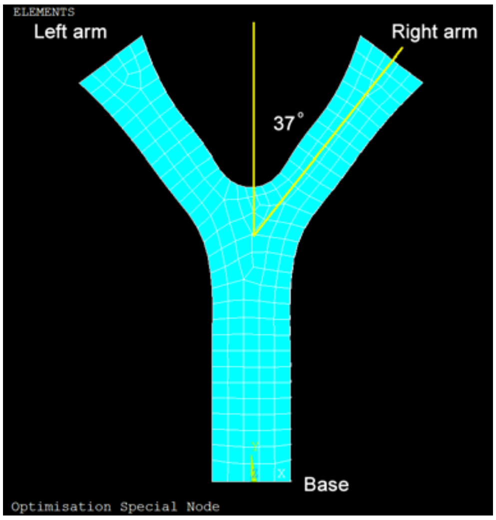


Figure 7.23. Preliminary design of node for FEM example case 1

The X- and Y-components of the shear-force in the node arms is calculated: (The angle of both the node arms $\alpha = 37^\circ$)

Y-component shear force left arm	= sin 37 * 10	= 6 kN	(negative y-direction)
X-component shear force left arm	= cos 37 * 10	= 8 kN	(negative x-direction)
Y-component shear force in right arm	= sin 37 * 15	= 9 kN	(negative y-direction)
X-component shear force in right arm	= cos 37 * 15	= 12 kN	(positive x-direction)

The X- and Y-components of the axial compressive force in both arms is calculated:

Y-component axial force left arm	= cos 37 * 800	= 640 kN (negative y-direction)
X-component axial force left arm	= sin 37 * 800	= 482 kN (positive x-direction)
Y-component axial force in right arm	=cos 37 * 1200	= 959 kN (negative y-direction)
X-component axial force in right arm	= sin 37 * 1200	= 722 kN (negative x-direction)

Now the reaction forces at support can be calculated:

Reaction force at support in Y-direction	= - 6 - 9 - 640 - 959	= -1614 kN
Reaction force at support in X-direction	= -8 + 12 + 482 - 722	= -236 kN

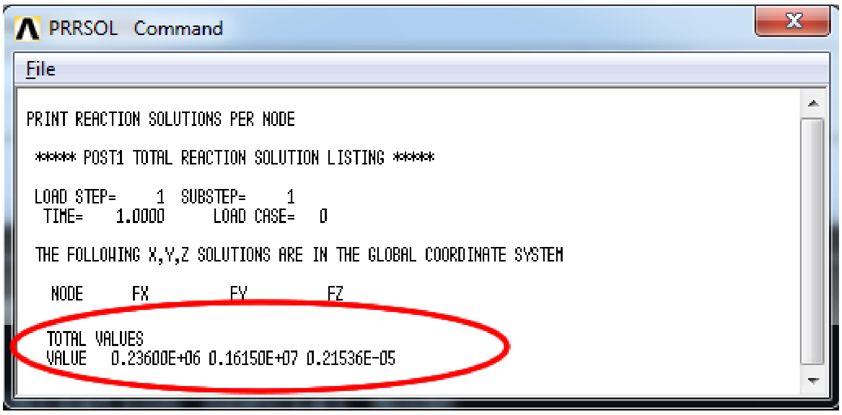


Figure 7.24. Reaction forces at support calculated by Ansys

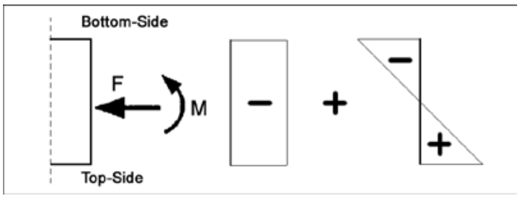
Now the hand calculations can be compared to the FEM calculation results. Ansys lists the reaction forces numerically and presents them in a table. See Figure 7.23. The values of the resulting reaction forces in both the X- and Y-directions from Figure 7.23 are similar to the resulting reaction forces calculated by hand. This verifies that the force distribution has been correctly transferred the model of the preliminary design. The FEM calculation results, with regard to the reaction forces, are correct and verified.

FEM example case 2: Stress Verification

In this test case, the stresses in the outermost fibres of the cross-section have been calculated in both the node arms. The calculations are fist made by hand and then followed by the Ansys calculation results.

The preliminary design and corresponding force distribution from example case 1 has been used for this example, see Figure 7.21 and 7.22. This particular preliminary design of the node has a rectangular cross-section with a varying width along the length of the node arms. The depth of the node is constant at 500 mm. The bending stress and axial compressive stress at the outermost fibre for the left arm is calculated by hand first and later compared to the stress output from Ansys.

The stresses in outer fibres of the left arm are:



Arm width =500 mm

Arm depth =500 mm

$A = 500 * 500 = 250000 \text{ mm}^2$

$W = \frac{1}{6} * 500^3 = 20{,}833 * 10^6 \text{ mm}^3$

$F = 800 \text{ kN}$

$M = 150 \text{ kNm}$

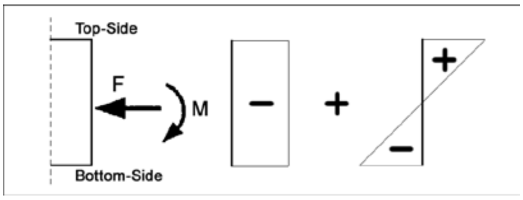
$\sigma_{compression} = 3.2 \text{ N/mm}^2$

$\sigma_{bending} = 7{,}2 \text{ N/mm}^2$

$\sigma_{bottom-side} = -10{,}4 \text{ N/mm}^2$

$\sigma_{top-side} = +4{,}0 \text{ N/mm}^2$

The same calculation is performed for the right arm:



Arm width =500 mm

Arm depth =500 mm

$A = 500 * 500 = 250000 \text{ mm}^2$

$W = \frac{1}{6} * 500^3 = 20{,}833 * 10^6 \text{ mm}^3$

$F = 1200 \text{ kN}$

$M = 80 \text{ kNm}$

$\sigma_{compression} = 4.8 \text{ N/mm}^2$

$\sigma_{bending} = 3{,}84 \text{ N/mm}^2$

$\sigma_{bottom-side} = -8{,}64 \text{ N/mm}^2$

$\sigma_{top-side} = -0{,}96 \text{ N/mm}^2$

The stresses obtained with Ansys are presented in Figures 7.24 to 7.30. In Figure 7.24 the 1st principal stress distribution (S1), which is the tensile stress, is displayed on the left and the 3rd principal stress-distribution (S3), which is the compressive stress, on the right. A positive value in the legend indicates a tensile stress and a negative value indicates a compressive stress.

For the left-arm, Figure 7.24 shows that the top-side has a tensile stress and the bottom-side has a compressive stress, as expected. For the right-arm, both the top-side and the bottom-side have only compressive stresses, which is also correct.

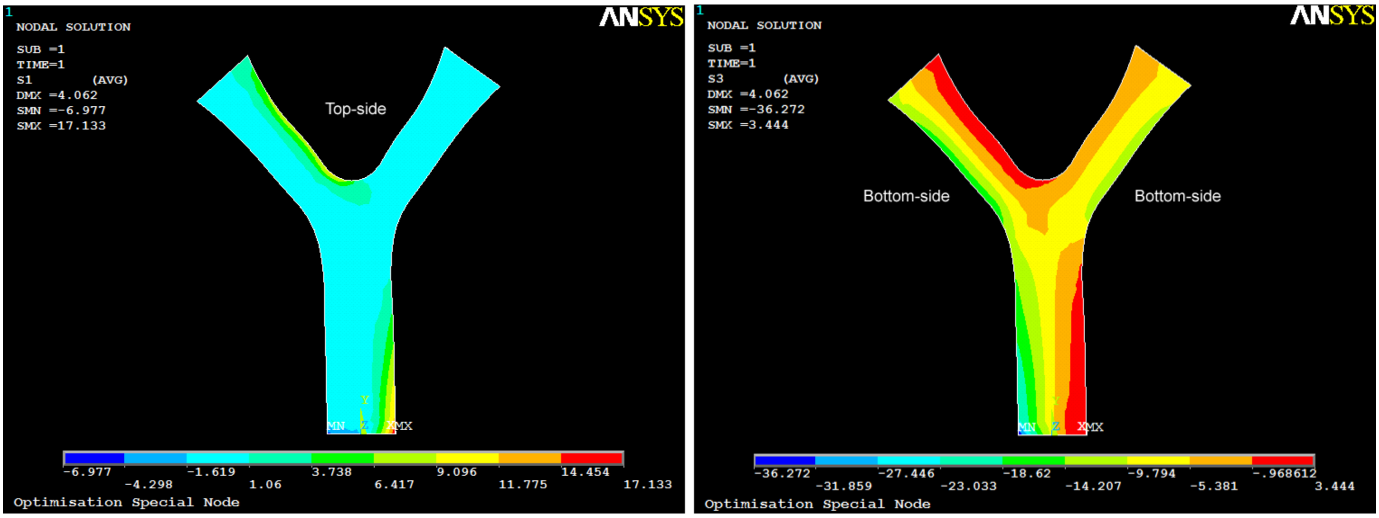


Figure 7.25. Principal stresses in the node. Left: tensile stress. Right: compressive stress

To check the exact magnitude of the local stresses in the **left-arm**, two finite-element-nodes on both sides are selected and examined. For the top-side the nodes 4573 to 4575 are selected. For the bottom-side the nodes 5853 to 5855 are selected. The location of the selected nodes and the corresponding stress is displayed in Figure 7.25 and 7.26.

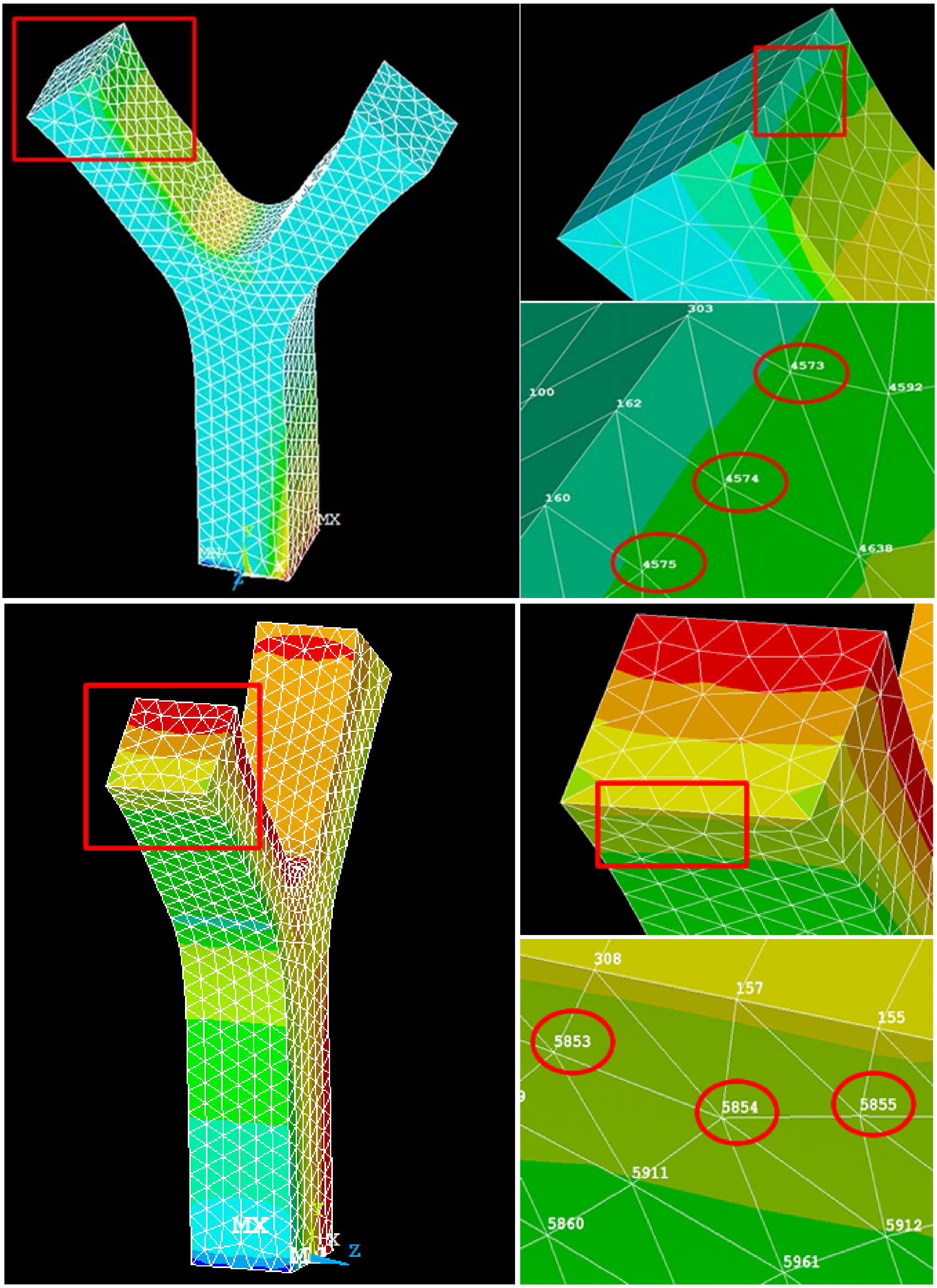


Figure 7.26. Finite-element-node locations in left-arm

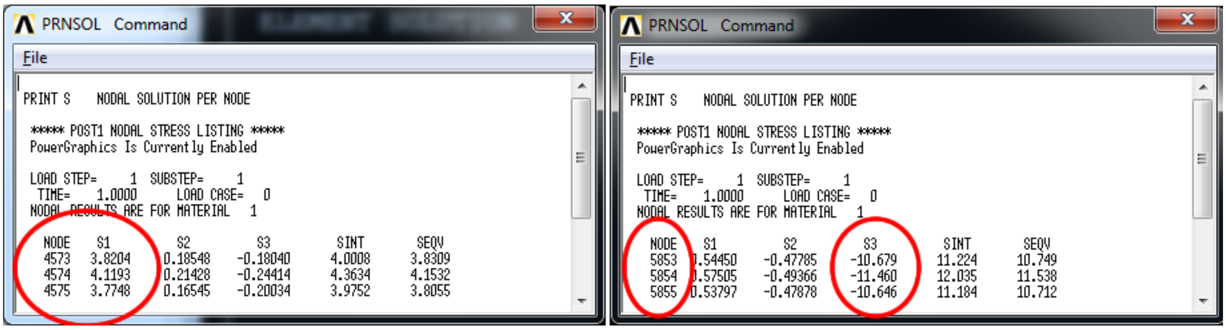


Figure 7.27. Finite-element-node stresses in left-arm

The stresses for the top-side are displayed on the left of Figure 7.26. When the three values for the tensile stress are averaged the resulting stress is: (3.82+4.11+3.77)/3 = 3.9 N/mm^2. The hand calculated value for the tensile stress at the top-side is 4 N/mm^2. The difference between the two is < 3%. This difference is acceptable.

The compressive stress at the bottom-side of the left-arm have an average value of: (10.67+11.46+10.64)/3=10.92 N/mm^2. The hand calculated value for the compressive stress at the bottom-side is 10,4 N/mm^2. The difference between the two is < 5%. This difference is also acceptable.

In order to check the magnitude of the local stresses in the right arm, nodes 4636 & 4637 are selected for the top-side. For the bottom-side the nodes 5152 to 5154 are selected. The location of the selected nodes and the corresponding stresses are displayed in Figure 7.27 and 7.28.

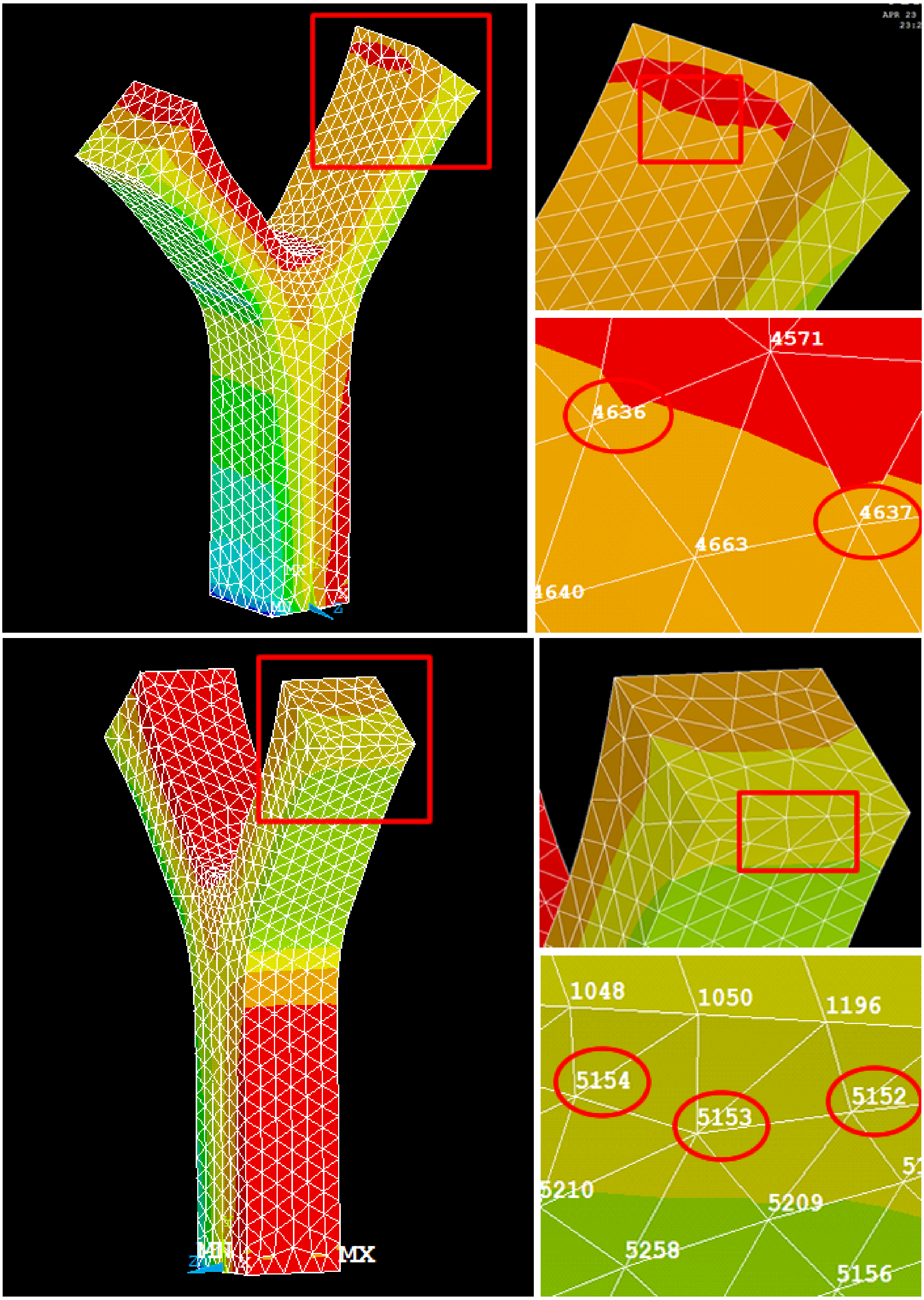


Figure 7.28. Finite-element-node locations in right-arm

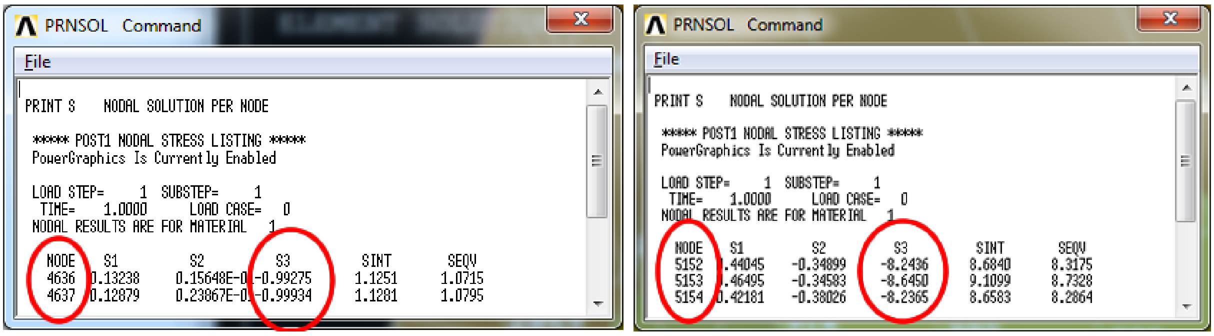


Figure 7.29. Finite-element-node stresses in right-arm

The stresses for the top-side are displayed on the left of Figure 7.28. When the two values for the compressive stress are averaged the resulting stress is: (0,99+0,99)/2 = 0.99 N/mm^2. The hand calculated value for the tensile stress at the top-side is 0,96 N/mm^2. The difference between the two is < 3%. This difference is acceptable.

The compressive stress at the bottom-side have an average value of : (8.24+8.64+8.23)/3= 8,37 N/mm^2. The hand calculated value for the compressive stress at the bottom-side is 8,64 N/mm^2. The difference between the two is < 3%. This difference is also acceptable.

All the stresses are within acceptable limits and the FEM stress calculations within *VisionNode* are considered to be correct.

FEM example case 3: Displacement verification

In this final test case, the displacement for the preliminary design of a node will be tested and evaluated. By inputting specific connection and dimensions properties into the graphical-user-interface, a rectangular rod can be created from within *VisionNode*. The input parameters are displayed in Figure 7.29 and the rod itself is displayed in Figure 7.30. Note the following properties for this element:

- UHPC mixture C170/200
- Young's Modulus 60000 Mpa
- Length of rod = 1000 mm
- Width of rod = 30 mm
- Depth of rod = 30 mm
- Width of left connection (where horizontal force is applied) = 2 mm
- Horizontal force at top = 1 kN

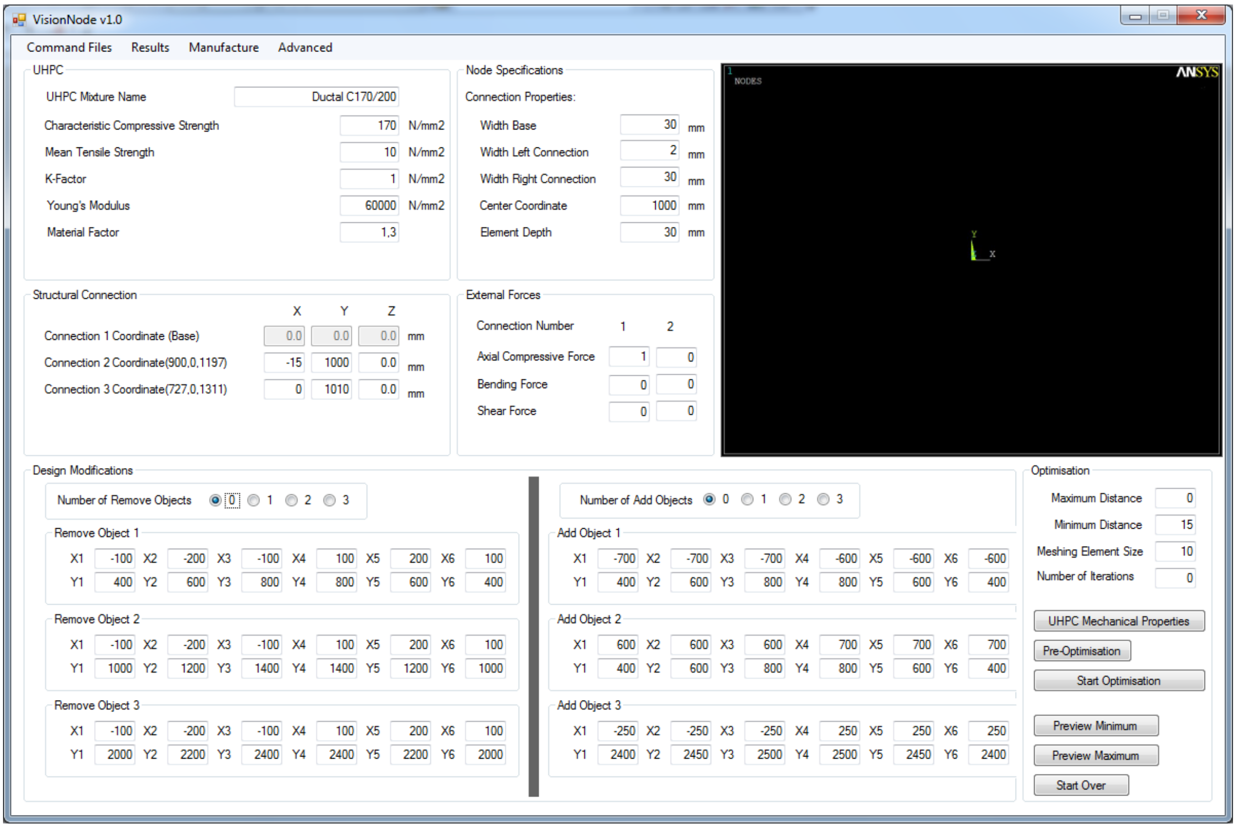


Figure 7.30. Input parameters for FEM example case 3

In Figure 7.30, the horizontal load on the rod is shown as Ansys displays it. In this figure the magnitude of the pressure, the horizontal force divided by the area, is indicated as 16,667 N. As was explained in Chapter 6, in *VisionNode* forces transferred from surrounding structure to the Special Node are expressed as surface pressure loads. Since the width of the left connection is 2 mm and the depth of the rod is 30 mm, the surface- area of the contact surface of the horizontal load is 2 * 30 mm= 60 mm^2. After dividing a force of 1kN through this area, a pressure value per mm^2 of 16,666 N is obtained. This is the same value of the pressure in Figure 7.30.

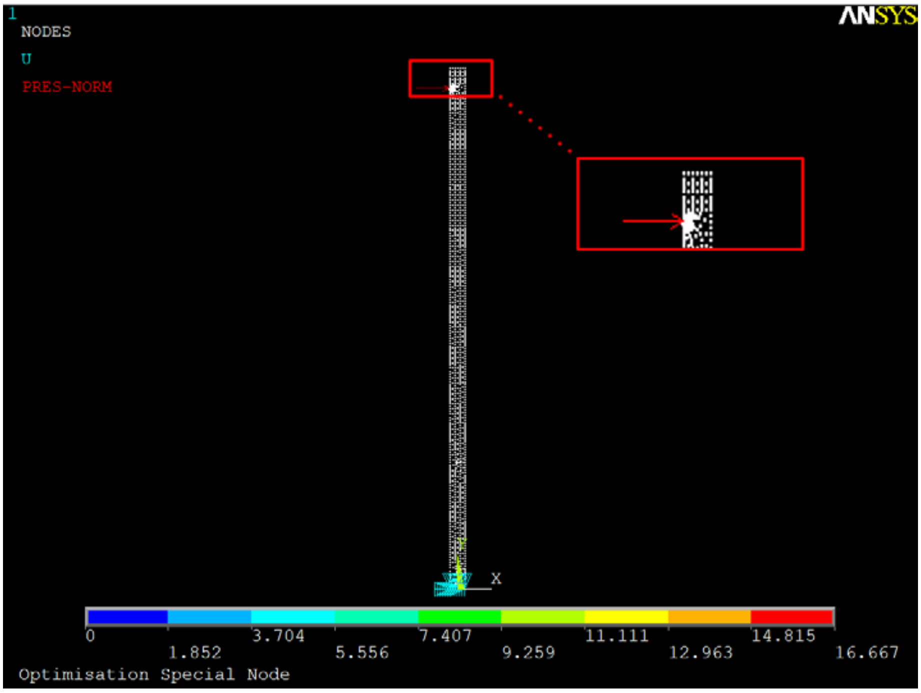
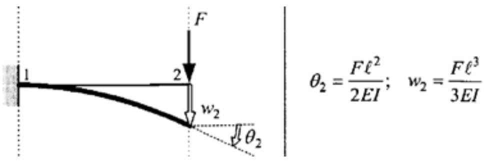


Figure 7.31. Rod and horizontal load on rod

The displacement at the top of the rod, can be calculated manually with a simple rule-of-thumb from structural-mechanics.



$$F = 1000 \text{ N}$$
$$E = 60000 \text{ MPa}$$
$$l = 1000 \text{ mm}$$
$$I = \frac{1}{12} * 30^4 = 67500 \text{ mm}^4$$

$$w = \frac{1000 * 1000^3}{3 * 67500 * 60000} = 82,3 \text{ mm}$$

The displacement calculated by Ansys is indicated by DMX, which essentially stands for maximum displacements. The DMX value is shown in Figure 7.31 and is:

DMX = 83,517 mm

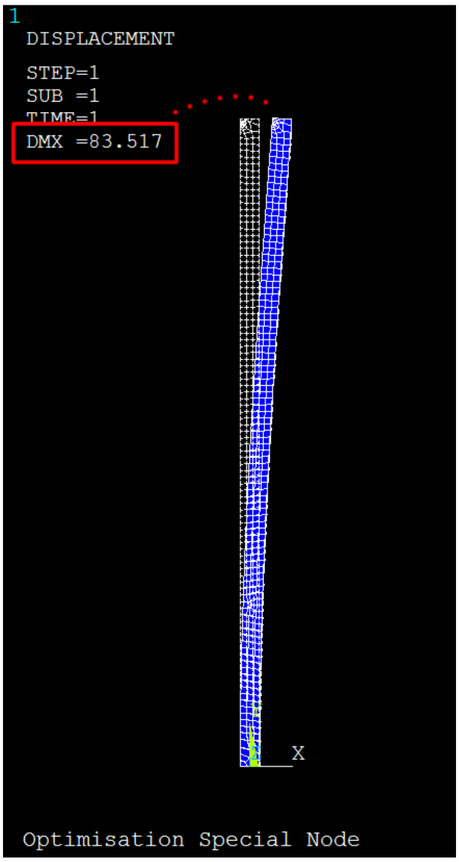


Figure 7.32. Displacement of the rod.

The difference between the hand calculated magnitude of the displacement and the Ansys calculated magnitude is < 2%. The difference is acceptable and can be attributed to meshing-element-size and pressure-load on a surface instead of a point-load.

From all the above FEM example cases it can be concluded that the FEM implemented in *VisionNode* performs correctly. FEM calculation results obtained from the complex calculations for the Special Nodes during optimisation, are also considered to be correct.

7.2.3. Optimisation Algorithm Evaluation

In this sub-section, the correct operation of the optimisation algorithm within *VisionNode* is tested and evaluated. The optimisation is powered by a genetic algorithm (GA). See Chapter 5 for more information on the GA. The example cases tested in this sub-section consists of an optimisation of a volume reduction.

Optimisation example case : Volume Reduction

This test is a fairly simple one. Using *VisionNode*, a preliminary design has been created without any remove- or add-objects. By utilising the Preview-Minimum functionality of the program, the absolute minimum value of the volume for this specific node can be calculated. Subsequently, the node is given a maximum geometry which serves as the upper-boundary for the optimisation design-variables. The maximum- and minimum geometry of this preliminary design is displayed in Figure 7.32 and the corresponding input parameters are shown in Figure 7.33. The volume of this particular preliminary design has been calculated by Rhinoceros and is displayed in Figure 7.32. Minimum volume of node: $1{,}014 * 10^9$ mm^3

Finally the optimisation has been run and its results compared to the known minimum value of the volume. The standard GA parameter-set from Figure 7.1 has been used for this optimisation. In Figure 7.34 the optimisation diagram is shown. The optimum value is displayed in the table below it. An optimum value of $1{,}055 * 10^9$ mm^3 was found. This value is < 4 % of the absolute minimum value of this particular preliminary design. This difference is acceptable.

Based on this result, the optimisation algorithm of *VisionNode* is considered to be functioning correctly.

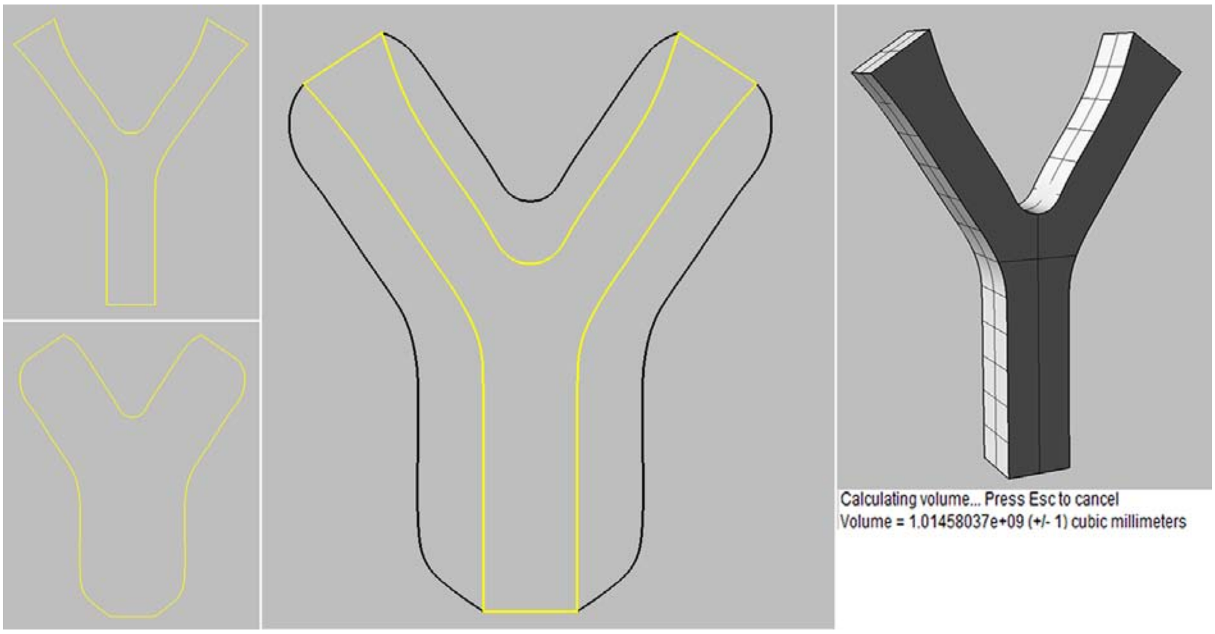


Figure 7.33. Preliminary design for optimisation example case

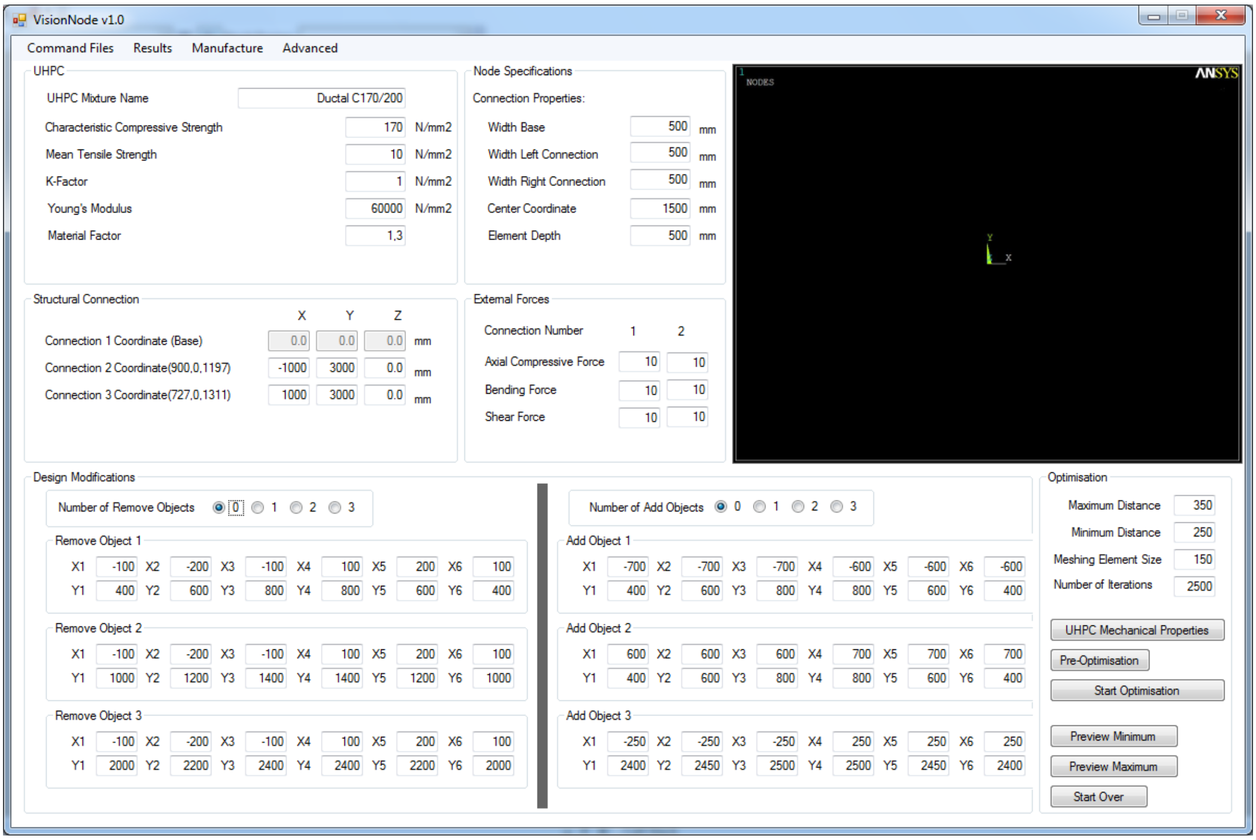


Figure 7.34. Input parameter for optimisation example case

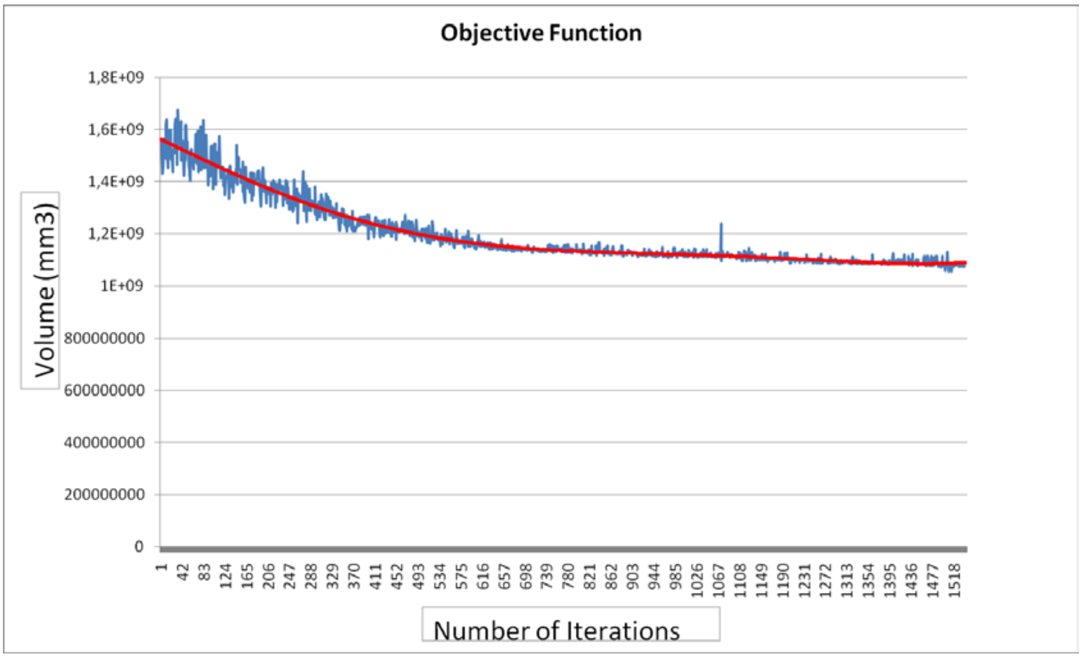


Figure 7.35. Objective Function of optimisation example case

Volume of Special Node (mm^3)	1055455896
Optimisation Time	1 hours

Table 7.8. Optimum value of optimisation example case

7.3. Conclusion

From the results of the operation of *VisionNode* and the evaluation presented in this chapter, the following conclusions and recommendations can be extracted.

Using *VisionNode* two Special Nodes in UHPC have been designed through computational optimisation. The genetic algorithm produced feasible designs of a Special Node that would not yield under the loads it was exposed to. For the first Special Node the UHPC mixture C170/200 was used successfully. It was not able to create the second Special Node using the same UHPC mixture because the tensile stresses were consistently higher the tensile capacity in the second Special Node. As a consequence, a different UHPC mixture with a higher tensile capacity was chosen. Using the new UHPC mixture, a feasible design of the second Special Node was created by *VisionNode*.

During the optimisations of the Special Nodes, a genetic algorithm parameters-set has been established and is referred to as the standard GA parameter-set. With this standard set, *VisionNode* was found to be able to optimise any preliminary design of a Special Node.

Through a number of example cases the software, Finite-Element-Model and the optimisation algorithm of *VisionNode* has been tested and evaluated. Considering the results of the evaluation, *VisionNode* has been found to function properly and correctly.

Chapter 8.

Manufacture

This research project has explored the possibilities of the technical feasibility of the creation of Special Nodes using UHPC with fibre-reinforcement and without passive reinforcement. To this purpose, a computational structural design tool has been developed in order to create the Special Nodes through computational optimisation. Furthermore, when looking at the results presented in Chapter 7, several Special Nodes have been successfully created using this tool for a specific example structure.

There remains, however, the matter of how to manufacture the digitally created Special Nodes and how to connect them to the surrounding structure. In addition, the type of connection has to be such that the forces between the Special Node and the structure are transferred as assumed in the applied structural calculation model.

In this chapter the manufacture of the Special Node is discussed in Section 8.1. In the subsequent section, the assembly and connection of the nodes to the surrounding structure is explained.

8.1. Manufacture Procedure

Using the computational structural design tool *VisionNode*, Special Nodes with a complex 3D-geometry can be created. When looking at the possibilities for the manufacture of such elements, one has to take several aspects into consideration. First of all, the construction material of the element itself partially determines what manufacture methods can be applied. Secondly, there is the consideration of creating the element on-site or by prefabrication. The maximum dimensions and transport also play a role.

Concrete elements are manufactured by pouring a specific concrete mixture in a mould where it is allowed to harden. After hardening the element is completed and ready to take part in the structural system. This manufacture process applies to all types of concrete and is no different for the Special Nodes in UHPC. In order to manufacture a Special Node, the following steps have to be undertaken. First, the mould for the Special Node must be created. When the mould is ready the UHPC mixture can be poured into the mould and allowed to harden. After sufficient hardening the UHPC can be cured by heat-treatment in order to increase its durability and strength properties. See Chapter 2 Section 2.2.1 for more information on heat-treatment. When the curing of the concrete is completed the Special Node is ready. In the case of off-site prefabrication, the Special Node must be transported to the construction site.

8.1.1. Mould manufacture

The Special Nodes designed in this research project are created through computational optimisation which involves the calculation of an optimum geometry. As a result, all the Special Nodes that are created have a complex geometry. A geometry can be considered complex when it involves combinations of double curved edges, varying angles between sub-elements and varying overall dimensions. This means that the mould for a Special Node must also have a complex geometry. Furthermore, the optimisation process determines all geometrical dimensions of the Special Node very precisely. For this reason, the geometrical dimensions of the mould must be very accurate.

Making complex geometrical moulds by hand using wood or steel, can be difficult and laborious. When the required geometrical accuracy of the mould is added on top of this, the

Figure 8.1. Fully automated 3D-milling machine (Wapperom,2004)

task becomes even more difficult. In order to facilitate the manufacture of complex geometrical moulds with accurate dimensions, the flexibility and precision of a machine is required. For this reason, the fully automated 3D-milling of the concrete mould using polystyrene as the mould material, is chosen as the method for the manufacture of the Special Nodes in this research project.

Fully automated 3D-milling of polystyrene is a method where a milling-machine is computer-controlled by a CAD/CAM-system to cut and mill polystyrene blocks into any desired geometry. Using CAD-software, such as Rhinoceros, the object geometry can be created digitally and afterwards used as input for the milling-machine computer. In turn, the machine creates a 3D-object based on the digital CAD-model. A 3D-milling machine is displayed in Figure 8.1.

3D-milling of polystyrene for the creation of concrete moulds has been used in the concrete industry for several decades now. One example is 'The Spencer Dock Bridge' in Dublin where multiple polystyrene blocks were created by 3D-milling and later assembled to form a single mould. After placing of the passive reinforcement bars, the concrete was poured and allowed to harden (www.AmandaLeveteArchitects.com). The result is displayed in Figure 8.2.

The mould material polystyrene, is a thermoplastic polymer material that is a solid substance at room temperatures. Pure polystyrene is a colourless, hard plastic with limited flexibility. Two of most applied adaptations of the material are extruded polystyrene (PS) and expanded polystyrene

Figure 8.2. The Spencer Dock Bridge in Dublin. Bottom-left: The polystyrene mould parts. Bottom-right: The finished section of the bridge (www.amamdalevetearchitects.com;

(EPS). Both of these are used in the concrete industry and are both suitable to serve as the mould material. The surface of the polystyrene mould, however, can be somewhat rough when a smooth surface is required. This can be remedied by applying a coating to the interior of the polystyrene mould. This way a concrete element with a smooth surface can be obtained (Wapperom, 2004). See Figure 8.3 for an impression of unprocessed polystyrene and a polystyrene mould created by milling.

Figure 8.3. Left: unprocessed EPS; Right: EPS concrete mould created by 3D-milling (Wapperom,2004)

Figure 8.4. Five-axis milling-machine (www.nedcam.nl).

Ongoing development in the milling-machine industry have produced multi-axis 3D-milling machines which can create more complex and accurate 3D-shapes (Wapperom, 2004). One of the newest types is the five-axis milling machine, which is displayed in Figure 8.4. This machine has become increasingly popular in the last few years because of its excellent milling capabilities and error minimisation. High precision of the machine allows for the smallest details in complex 3D-geometry (Makhanov & Anotaipaiboon, 2007). The largest of its kind in Europe, is located in the Netherlands and has the capability to mill objects with a maximum dimension of 10 by 3,6 by 3 meters. Such a milling machine would be required for the mould manufacture of the Special Nodes.

For a single Special Node, two polystyrene moulds would have to be made because a Special Node always contains holes. For this reason, two halves of the same mould must be created to effectively manufacture the Special Node. To give an idea how a polystyrene mould of a Special Node would look like, this concept is illustrated in Figure 8.5 to 8.7. The Special Node in Figure 8.5 will require two of the moulds displayed in Figure 8.6. The contours of the node are highlighted in yellow in this figure. When the two halves are joined together, the mould for the Special Node is complete. See Figure 8.7. The mould for any Special Node would have to be created as such.

The models displayed in Figure 8.5 to 8.7 are CAD-models created with Rhinoceros which saves its models with the file-extension *.3DM. A 3DM-file is directly recognised and accepted by a 3D-milling machine computer. This means that in theory, the displayed models can actually be created at this time. This goes for all Special Nodes created by *VisionNode* because after optimisation, the model is saved as a 3DM-file.

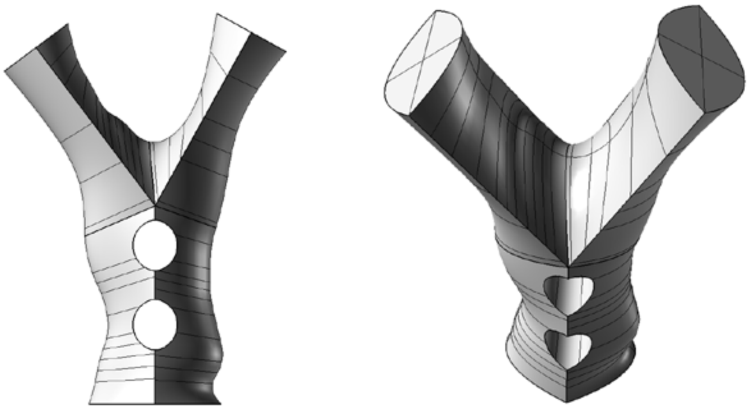


Figure 8.5. A Special Node created by VisionNode

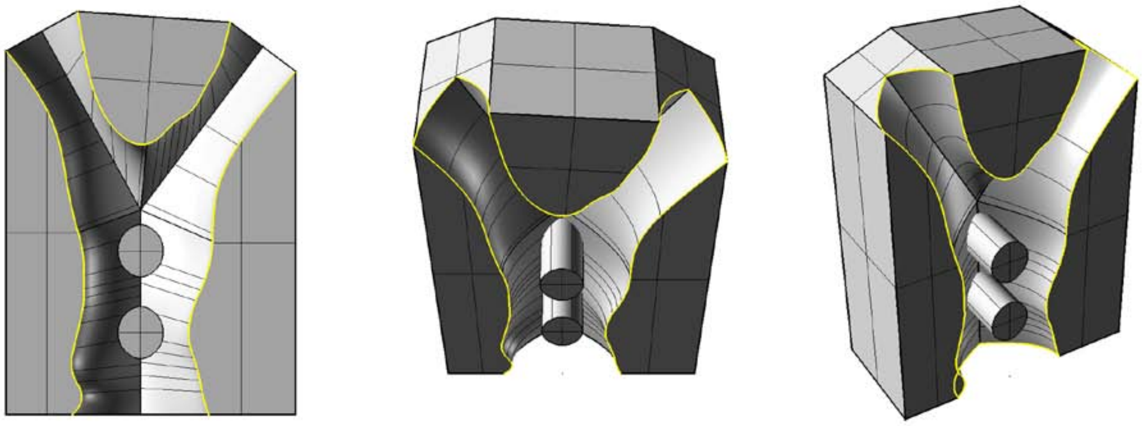


Figure 8.6. One halve of the mould needed for manufacturing the Special Node

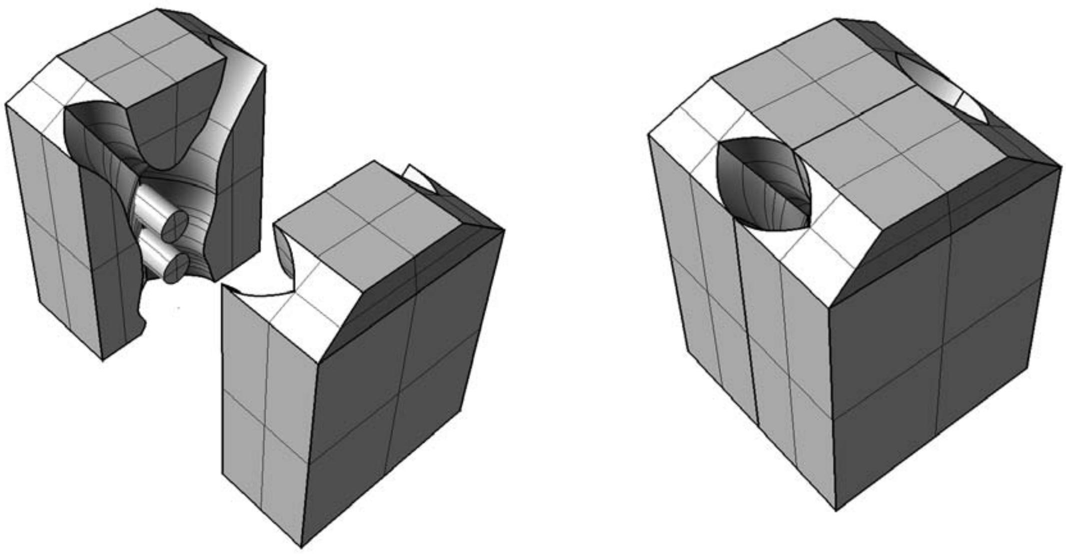


Figure 8.7. Two mould halves for a single mould for the Special Node

8.1.2. Special Node Manufacture

Now that the mould manufacture for the Special Node has been explained, the next step is to pour the UHPC mixture into the mould and allow it to harden. UHPC is a material for which constant supervision and monitoring during hardening is advised in order for it to obtain the characteristic design strengths (Goldbach & Stehling, 2008). Because of this, the Special Nodes would have to be prefabricated in a controlled environment and cannot be made on site. After fabrication they can be transported to the construction site for assembly.

The tensile behaviour of the cast UHPC element is determined by the fibre orientation in the mixture. The fibre orientation can be controlled, to a certain level, by the direction and distance of pouring the UHPC mixture into the mould (Pansuk et. al, 2008). For this reason, the pouring process must be supervised and the pouring direction must be adjusted to every specific Special Node. Since, *VisionNode* produces a finite-element-model (FEM) for each Special Node, the location of the large tensile stresses can be established. As such, the pouring of the UHPC mixture can be modified accordingly. After the Special Nodes has hardened, the element must be heat-treated in order to improve durability and strength properties. This must also be done in a controlled environment.

The final step is to transport the element to the construction site and connected to the surrounding structure. This is discussed in the next section.

8.2. Connection & Assembly

In this research project, the Yas-Hotel structure in Abu-Dhabi serves as the example structure for which a Special Node has been designed using *VisionNode*. The connection between the Special Node and the diagonal steel members of the Yas-Hotel structure, is a steel-UHPC connection. As such, properties of this connection have to be explained. In this section, one alternative for the structural steel-UHPC connection between the Special Node and the diagonal steel members is presented. Subsequently, an example connection between a Special Node and a concrete element is discussed in order to show how such a connection could be made, even though this is not needed for the Yas-Hotel structure.

The FEM model that is implemented in *VisionNode* is discussed in Chapter 4 of this report. It assumes a certain transfer of forces between the Special Node and the connection elements. For this reason, the type of the structural connection between a Special Node and the connection element must be such, that the assumed force transfer in the FEM model can be substantiated.

8.2.1. Steel-UHPC connection

To facilitate a steel-UHPC connection between the Special Node and the diagonal steel elements of the Yas-Hotel structure, a steel head-plate can be used. See Figures 8.8 and 8.9. During casting the steel head-plates is anchored in the UHPC of the Special Node. After hardening the steel head-plate and the UHPC form one structural system.

In Chapter 6 it was explained that the compressive force from the structure is transferred to the Special Node as a distributed pressure load. This head-plate can facilitate this pressure load as it covers the entire contact surface between Special Node and structure. The steel plate also has the needed stiffness to transfer the force as described. The same applies for the bending moment and shear force, that are distributed along the contact surface because of the steel plate.

Additionally, the diagonal steel members can be welded or bolted to the steel head-plate. In case of a bolted connection, the head-plate must be equipped with bolting holes. Assembly tolerances must also be considered during construction.

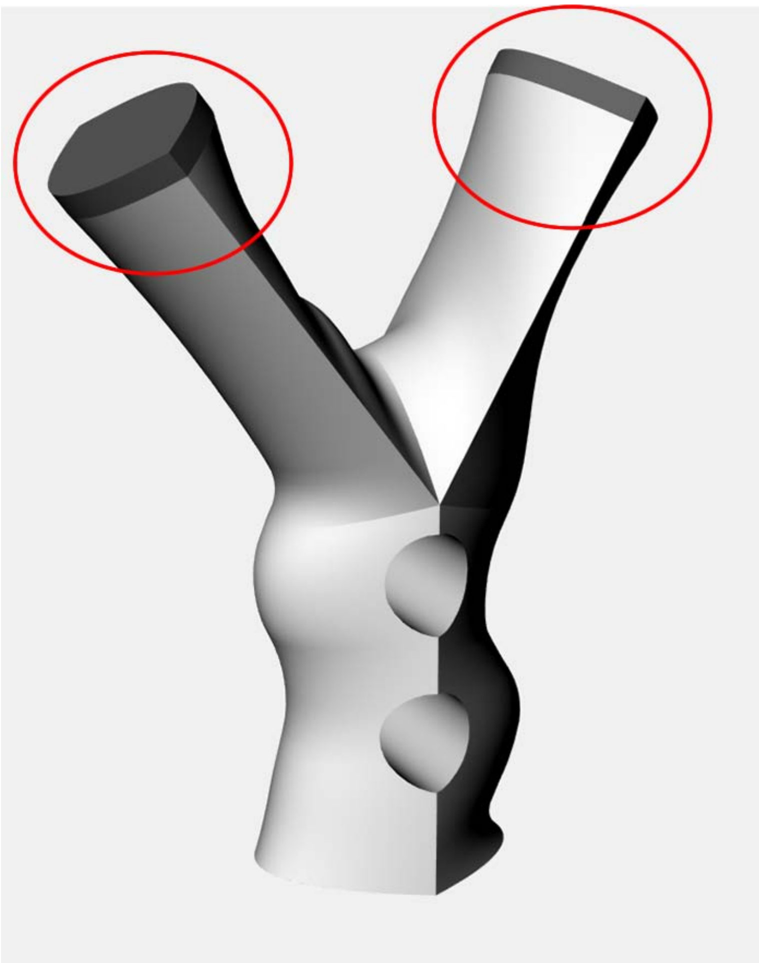


Figure 8.8. Special Node with a steel head-plate for a steel-UHPC connection

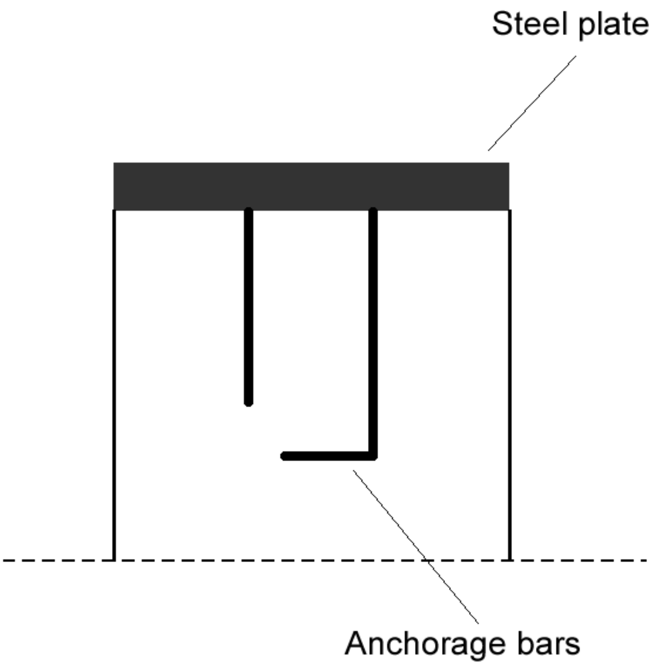


Figure 8.9. The anchoring of the steel head-plate in the UHPC

8.2.2. Concrete-UHPC connection

In the case of a concrete-UHPC connection, the alternative presented in Figure 8.10 and 8.11 can be used. Steel bars are anchored in the Special Node during casting to which a concrete element can be fastened. For prefabricated concrete elements a grouted connection can be used. For in-situ concrete, appropriate formwork has to be build around the contact surface of the Special Node after which the concrete is cast.

Since there are only compressive axial forces and no pure tensile axial forces transferred from structure to Special Node, this alternative is adequate.

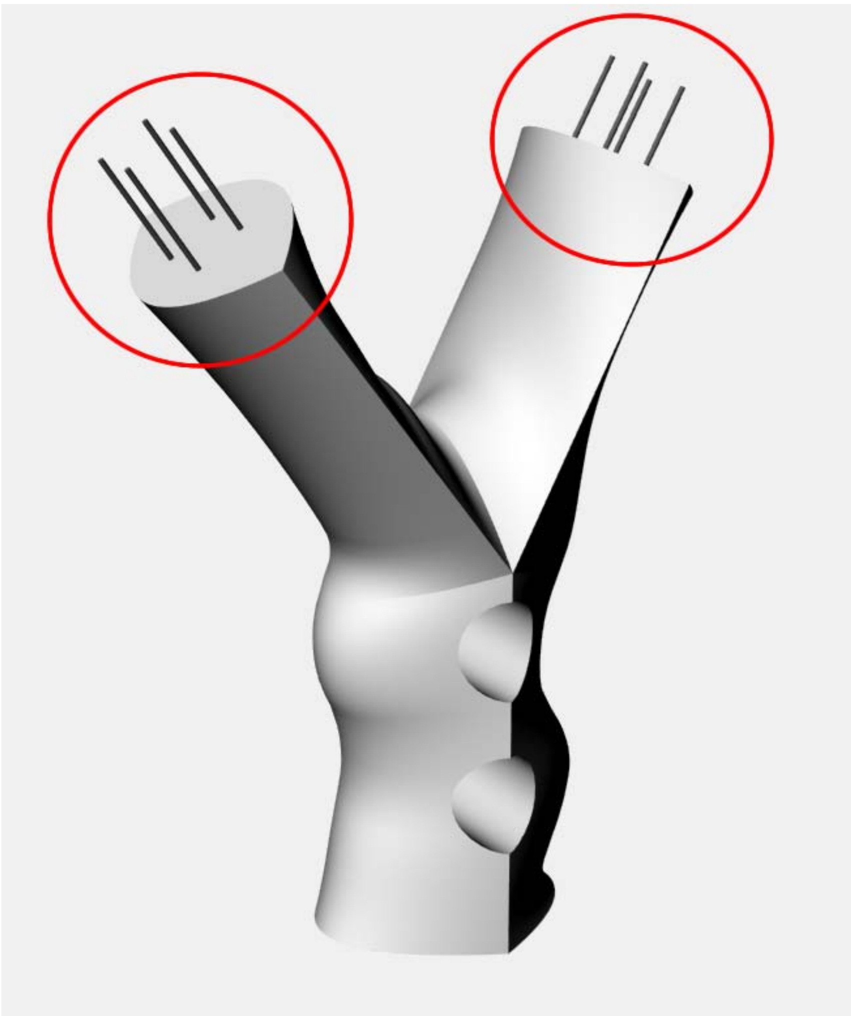


Figure 8.10. The Special Node with steel bars for a concrete-UHPC connection

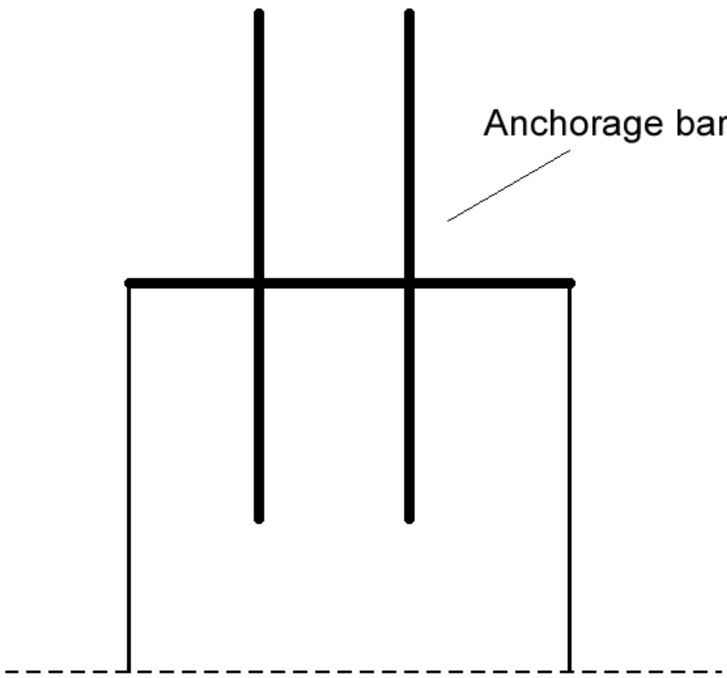


Figure 8.11. The anchorage of the steel bars in the UHPC

8.3. Conclusions

In this chapter, the manufacture and the assembly of the Special Nodes has been discussed. To manufacture a Special Node, first a mould must be created using fully automated 3D-milling of polystyrene as the mould material. The final Rhinoceros 3D-model of the Special Node is saved with the file-extension *.3dm and can be directly transferred to the computer of the 3D-milling machine.

For every Special Node, two moulds must be created because a Special Node always contains holes. The manufactured mould must then be used to cast the Special Node in UHPC. During casting, a steel head-plate can be anchored in the UHPC to facilitate a steel-UHPC connection. In the case of a concrete-UHPC connection, steel bars can be anchored in the UHPC.

After the Special Node has been cast, it must be cured by heat-treatment. When this is complete the Special Node is ready and can be transported to the construction site for assembly.

Chapter 9.

Conclusions & Recommendations

In this chapter the conclusion based on the outcome of this research project are formulated. Additionally, several recommendations are stated based on the conclusions. All research conclusions are theoretical since no lab experiments have been performed.

9.1. Conclusions

For every research objective that have been presented in Chapter 1, a conclusion is formulated in this section.

Conclusion 1

> *" In this research project, it has been determined that it is technically feasible to create Special Nodes, as defined in this research, using Ultra High Performance Concrete with fibre reinforcement and without passive reinforcement "*

Regarding the first conclusion, several additional observations were made and secondary conclusions can be drawn from these:

- The direct tensile capacity of the applied UHPC mixture is the governing factor that determines whether a Special Node for a specific design is technically feasible. When designing Special Nodes with multiple holes, a UHPC mixture with a high tensile strength must be used
- The compressive strength of the UHPC mixture is not the governing factor that determines whether a Special Node for a specific design is technically feasible. However, the compressive capacity is part of the Mohr-Coulomb-Failure-Check used and must be high enough
- Because the fibre-orientation partially determines the tensile capacity of a UHPC mixture, the orientation of the fibres must be adequate at locations in the Special Node with high tensile stresses. These locations can be obtained from the Finite-Element-Model of the Special Node

Conclusion 2

> *" A fully operational computational structural design tool called VisionNode, that creates Special Nodes using computational optimisation, has been developed and performs correctly "*

Regarding the second conclusion, the following observations have been made:

- For the genetic algorithm, a standard genetic algorithm parameter-set has been established which produces optimised designs of a Special Node for every preliminary design
- The implemented key features have a profound positive effect on the operation and efficiency of *VisionNode* and the Special Nodes it produces

Conclusion 3

" The user of the computational structural design tool is accommodated by the tool to influence the design of the Special Node. The user has the ability to add and remove material at any desired location in the Special Node "

Regarding the third conclusion, the following is stated:

- The Special Nodes, that can be designed using the current version of *VisionNode*, have three node arms and therefore can be connected to three incoming structural members
- Using the interactive graphical-user-interface of *VisionNode*, users can make a preliminary design of a Special Node. This preliminary design contains all design preferences and design modifications specified by the user. *VisionNode* creates a Special Node based on this preliminary design.

9.2. Recommendations

Based on the results and conclusions of this research, the following recommendations are made:

- The perspectives of optimisation and design possibilities defined in this research project, are not limited to structural nodes only. The methods and the techniques used in this research can and should be applied to other structural elements as well
- Even though computational optimisation is being used by *VisionNode* to create Special Node, the overall design process remains an iterative one. It is advisable to execute several optimisation runs of each preliminary design to have multiple optimum solutions and optimum design to chose from
- The Special Nodes should actually be manufactured and experimentally tested. *VisionNode* supplies all the data and models that are required as input
- The functions and features of the computational structural design tool *VisionNode* that has been developed in this research, should be expanded with additional design possibilities such as:
    - Additional Add- and Remove-Objects
    - Additional node arms
    - Connection possibilities for connections in two planes

- The time that is required for the optimisation process, is mainly determined by the FEM calculations. The speed of the FEM calculations is determined by the write/read-speed of the hard-drive in the computer. Faster hard-drives should be used to speed up the optimisation process. When SSD's are used in RAID-0 configuration, the optimisation time can theoretically be reduced by a factor 10

9.3. Finalising this research project

Based on the conclusions of this research project, it has been shown that computational optimisation combined with relatively new materials and manufacture methods can be used to introduce new design possibilities and applications for structural elements. By applying computational optimisation to seemingly illogical and inefficient preliminary design of structural nodes, unique structural nodes can be created which are technically feasible at the same time.
As was stated in the recommendations, this should not be limited to structural nodes and the same paradigm should be used for other structural elements as well.

To finalise this research project, the structural node that was manually created in Chapter 3 is placed next to the Special Nodes produced by *VisionNode*. See Figure 9.1 on the next page. Both nodes have been designed for the Yas-Hotel structure. The purpose of this comparison is not to prove that Special Nodes look better or are more efficient that the manual design, but to show that new design possibilities for this structural elements are real and can be realised today. Conventional methods of design and construction should not be changed. However, new applications and possibilities should not be overlooked either.

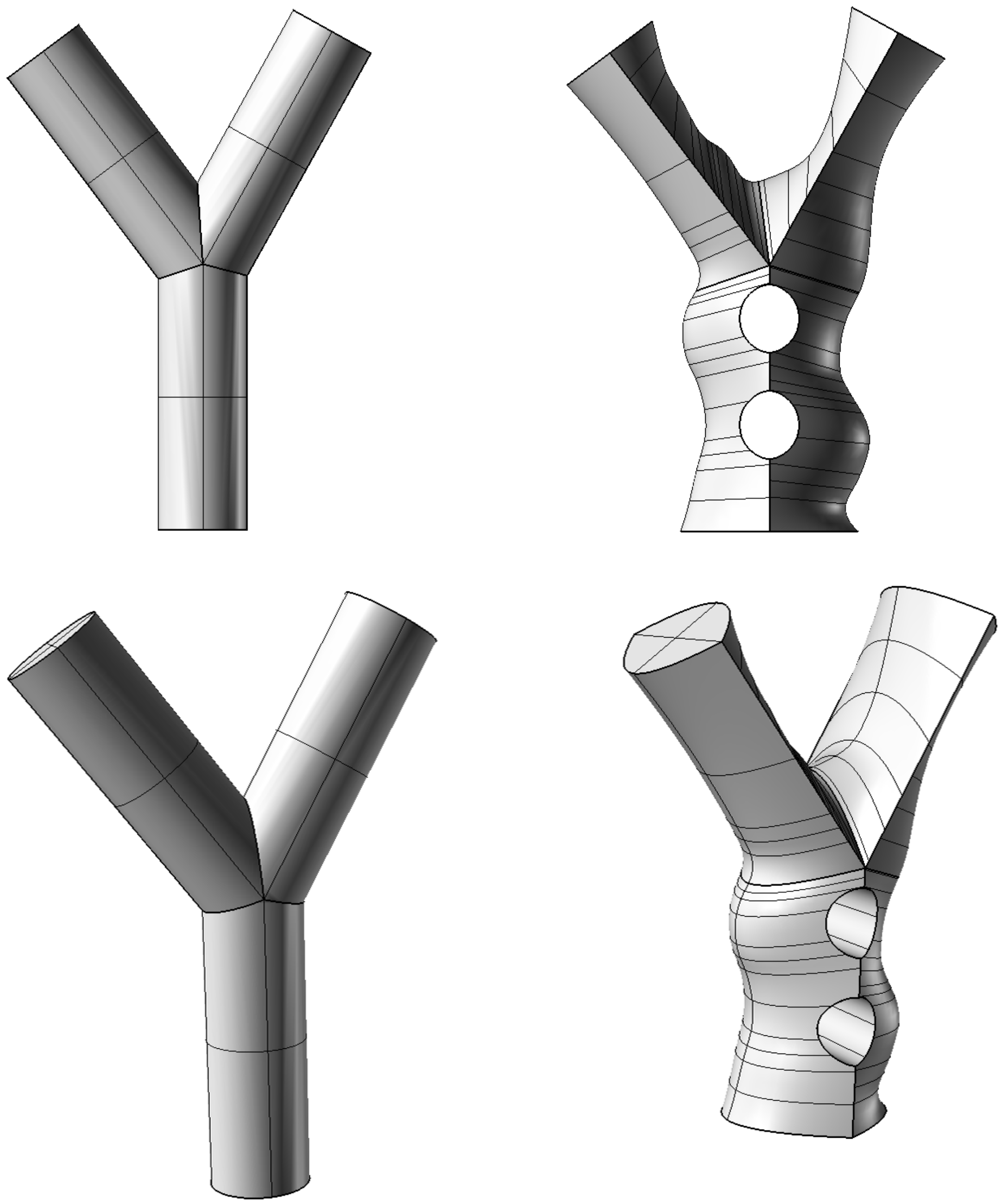


Figure 9.1. Two different designs of a structural node for the Yas-Hotel building. Left: manually designed structural node. Right: Special Node

Appendix

Appendix A

Structural buckling check

The buckling check is performed on the diagonal steel member of the model of the Yas Hotel structure. In Figure A.1 the member is presented. The left member in the figure is to be checked on buckling.

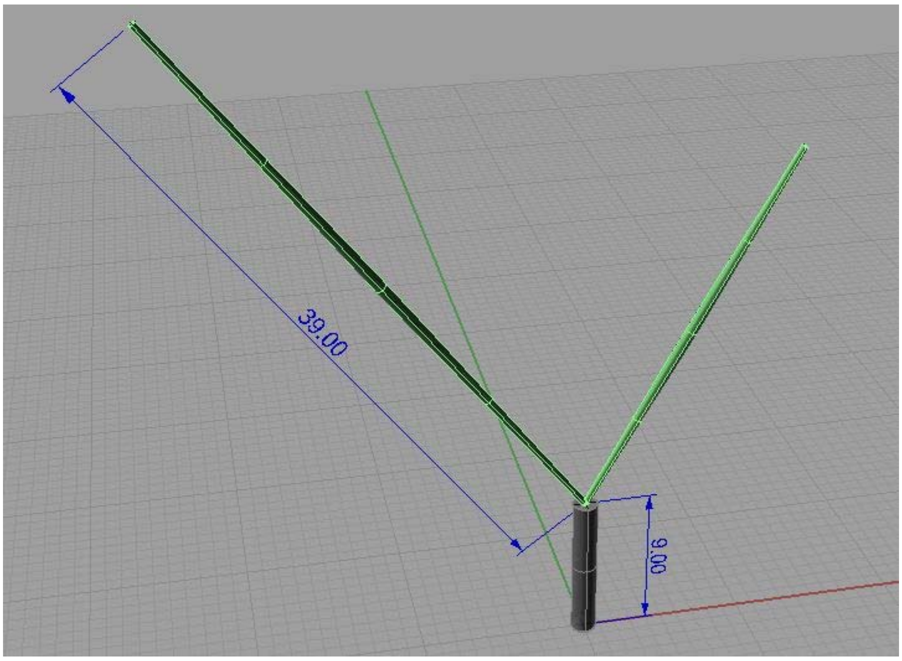


Figure A.1. Diagonal steel members. Units: meters

In Figure A.2 the normal force on the members is presented. The normal force is calculated in the GSA model based on loads on the structure portrayed in Section 3.4 of Chapter 3.

Figure A.2. Normal force on member. Unit: kN

Applied code: Eurocode 3: 1993 1-1. Article 6.3.

Cross-sectional properties of diagonal steel member:

Length = 39000 mm
Buckling length= 39000 mm
Cross-sectional class 3
Buckling-curve a

$A = 21677 \text{ mm}^2$
$I_{zz} = 1290 * 10^6 \text{ mm}^4$
$i = \sqrt{\frac{Izz}{A}} = 244 \text{ mm}$

Buckling check formula:

$$\frac{N_{Ed}}{N_{b,Rd}} \leq 1,0$$

N_{Ed} = Compressive force in element = 1488 kN
$N_{b,Rd}$ = Buckling resistance of element

$$N_{b,Rd} = \frac{\chi * A * f_y}{\gamma_{M1}}$$

$\chi =$ Reduction-factor taken from buckling curve using relative slenderness $\overline{\lambda}$
$A =$ Cross-section element
$f_y = 235 \text{ N/mm}^2$
$\gamma_{M1} = 1,00$ for buildings

Relative slenderness $\overline{\lambda}$
$$\overline{\lambda} = \frac{L_{cr}}{i} * \frac{1}{\lambda_1}$$
$L_{cr} = 39000 \text{ mm}$

$$\lambda_1 = \pi\sqrt{\frac{E}{f_y}} = \pi\sqrt{\frac{210000}{235}} = 94$$

$$\bar{\lambda} = \frac{39000}{244} * \frac{1}{94} = 1,7$$

The corresponding reduction-factor χ =0,3

The buckling-check:

$$\frac{N_{Ed}}{N_{b,Rd}} \leq 1,0$$

$$\frac{1488}{\left(\frac{\chi * 21677 * 235}{1,00}\right)} = 0,97$$

<u>The buckling-check is satisfied</u>

References

Acker, P., Behloul, M., 2004, Ductal® Technology: A Large Spectrum of Properties, A Wide Range of Applications. fib Symposium, Avignon, France

Adams B.M. et al., 2009. DAKOTA, A Multilevel Parallel Object-Oriented Framework for Design Optimisation, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis. Users's Manual. Unlimited Release, Sandia International Laboratories.

Beasley, D et al. 1993, An overview of Genetic Algorithms, part 1: Fundamentals, University Computing, pp. 58-69

Budd, T. 1996. An introduction to Object-Oriented programming. Addison-Wesley Publishing (Sd).

Cwirzen, A., Habemehl-Cwirzen, K. and Penttala, V., 2008. The effect of heat treatment on the salt freeze-thaw durability of UHPC. *Proceedings of the International Symposium on Ultra High Performance Concrete*, Kassel University Press, pp. 221-230

Geisenhanslüke, C. and Schmidt, M., 2004. Modeling and Calculation of High Density Packing of Cement and Fillers in UHPC. *Proceedings of the International Symposium on Ultra High Performance Concrete*, Kassel University Press, pp. 303-312.

Goldbach, U., Stehling, S., 2008, Precasting of UHPC elements. *Proceedings of the Second International Symposium on Ultra High Performance Concrete,* Kassel University Press, pp. 589-595.

Graybeal, B.A., 2006. *Material property characterization of Ultra High Performance Concrete.* FHWA-HRT-06-103, Federal Highway Administration, U.S. Department of Transportation.

Hogere sterkte beton, Ervaring met sterkteklasse B65 en hoger. 1999. Report BSW 99-20, Bouwdienst Rijkswaterstaat, Utrecht.

Kaptijn, N., 2002. Toekomstige ontwikkelingen van zeer-hogesterktebeton, De visie van Rijkswaterstaat. Cement nr. 2 , pp. 56 - 63.

Kingsley-Hughes, A. 2007. C-Sharp 2005 Programmer's Reference. Wrox press.

Maeder, U. et al., 2004, Ceracem, a new high performance concrete, characterisations and applications. *Proceedings of the International Symposium on Ultra High Performance Concrete*, Kassel University Press, pp. 59-68.

Makhanov,S.S., Anotaipaiboon, W. 2007. Advaned Numerical Methods to Optimize Cutting Operations of Five Axis Milling Machnines. Springer, 206p.

Michell, A., 1904. The limits of economy of material in frame-structures. Phil. Mag. pp. 589-597.

Pansuk W. et al, 2008, Tensile behaviour and fibre orientation of UHPC. *Proceedings of the Second International Symposium on Ultra High Performance Concrete,* Kassel University Press, pp. 161-168.

Peters, H. et al, 2008, Durable adhesive bonding with epoxy resins in civil engineering construction. *Proceedings of the Second International Symposium on Ultra High Performance Concrete,* Kassel University Press, pp. 267-274.

Rebentrost, M. and Wight, G. 2004, Experience and Applications of Ultra High Performance Concrete in Asia. *Proceedings of the International Symposium on Ultra High Performance Concrete*, Kassel University Press, pp. 19-30.

Resplendino, J. 2004. Firs Recommendations for Ultra High Performance Concrete and examples of Applications. *Proceedings of the International Symposium on Ultra High Performance Concrete*, Kassel University Press, pp. 79-90.

Resplendino, J. and Petitjean, J., 2003. Ultra High Performance Concrete: First recommendations and examples of application. ISHPC 1.

Richard, P. and Cheyrezy, M., 1995. Composition of Reactive Powder Concrete Research. Cement and Concrete Research, 25(7):pp. 1501-1511.

Schmidt, M. and Fehling, E., 2007. Ultra-High-Performance Concrete: Research, Development and Application in Europe. *10 Years of research and development at the University of Kassel*, pp. 194-221.

Schutter, G. de and Apers, J., 2007. Hoogwaardig beton. FEBELCEM Dossier Cement, Nummer 40.

Teichmann, T. and Schmidt, M., 2004. Influence of the packing density of fine particles of structure, strength and durability of UHPC. *Proceedings of the International Symposium on Ultra High Performance Concrete*, Kassel University Press, pp. 313 - 323.

Theory reference for Ansys and Ansys Workbench, 2007. Ansys Inc.

Veendendaal, D., 2007. Preliminary study on Evolutionary Optimisation of Fabric Formed Structural Elements. Delft University of Technology.

Wuest, J. et al, 2008, Model for predicting the UHPFRC tensile hardening response. *Proceedings of the Second International Symposium on Ultra High Performance Concrete,* Kassel University Press, pp. 153-160.

Yas Hotel Press Release, 2009. Retrieved August 2009: http://www.asymptote-architecture.com

Zimmermann, S., 2002. Nieuw composietmodel voor de elasticiteitsmodulus van hogesterktebeton, Een structuurgeoriënteerd praktijkgericht model. Cement, 6. pp. 99 - 106.

Building codes

AFGC / SETRA ,2002. Ultra High Performance Fibre-Reinforced Concretes: Interim Recommendations.

CUR Aanb. 97, *Aanbeveling 97, Hogesterktebeton,* Civieltechnisch Centrum Uitvoering Research en Regelgeving, Gouda.

DAfStB UHPC, 2003. State-of-the-art Report on Ultra High Performance Concrete-Concrete Technology and Design. Deutscher Ausschluss fur Stahlbeton/German Association for Reinforced Concrete, Berlin, draft 3.

Eurocode 1 : 1991

Eurocode 3 : 1993

Japanese Society of Civil Engineers,2008. Recommendations for Design and Construction of High Performance Fibre Reinforced Cement Composites with multiple fine Cracks (HPFRCC).

NEN6720, 1995, Voorschriften beton, TGB 1990, Constructieve eisen en rekenmethoden (VBC 1995), 2nd ed., Nederlands Normalisatie-instituut, Delft.

Internet

http://www.bekaert.com, August 2009

http://www.asymptote-architecture.com, August 2009

http://www.kpf.com, August 2009

http://www.c-i-d.dk, August 2009

http://www.ductal-lafarge.com, August 2009

http://www.imagineductal.com, August 2009

http://www.nedcam.nl, May 2010

http://www.delaroyepsdesign.nl, May 2010

http://www.amandalevetearchitects.com, May 2010

http://www.sandia.org