DEVELOPMENT OF 3D MODELING METHODS OF CRUSHER WEAR PARTS

Lappeenranta–Lahti University of Technology LUT

Master’s Programme in Mechanical Engineering, Master's thesis

2024

Mohammad Reza Heshmati

Examiners: Professor Jussi Sopanen

Charles Nutakor , D.Sc. (Tech.)

ABSTRACT

Lappeenranta–Lahti University of Technology LUT
LUT School of Energy Systems
Mechanical Engineering

Mohammad Reza Heshmati

Development of 3D modeling methods of crusher wear parts

Master’s thesis
2024
57 pages, 35 figures, 4 tables
Examiners: Professor Jussi Sopanen and Charles Nutakor , D.Sc. (Tech.)

Keywords: Generative design, algorithmic feature, algorithmic design, 3D modeling, parametric design, complex pattern, patterning feature

This thesis explores using the algorithmic feature in generative design for the 3D modeling development with complex patterned inserts like those found in the Metso crusher wear parts. The modeling method includes predetermined specifications for insert features, like length and diameter. The logic editor uses a variety of nodes to establish design criteria, such as the distance between inserts, the amount of back thickness left, and the surface area of each insert. These methods include the use of grids, lists, equations, and conditions. Comparing algorithmic and manual modeling techniques produces promising results, especially when using the algorithmic feature. This creative approach not only saves the organization a great deal of time and money but also gives designers a chance to improve their algorithm design skills. It also makes it possible to apply this strategy to different models with different sizes and geometries but with comparable design objectives.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor, Professor Jussi Sopanen, and Charles Nutakor for their invaluable guidance and support throughout the process of writing this thesis. Their insightful feedback and encouragement during our monthly meetings have been instrumental in shaping the progress and refinement of this work.

I am deeply thankful to Mikko Malkamäki and Aaron Baigent from Metso Company for providing me with the opportunity to undertake this project and for their unwavering assistance and support throughout its development.

I extend my heartfelt appreciation to Nicolas Hamilton for his invaluable contributions during the brainstorming sessions for this thesis and his continuous support. Additionally, I am grateful to the experts from Siemens who assisted in exploring new methods of design.

Lastly, I am profoundly grateful to my family, especially my mother and father, for their unwavering support and encouragement. Their belief in me has been the driving force behind this significant milestone in my life journey.

Mohammad Reza Heshmati

Tampere, Metso 21-03-2024

SYMBOLS AND ABBREVIATIONS

Roman characters

C	Circumference
d	diameter
L	length
l	insert length
r	radius
W	width

Greek characters

α	remaining thickness between inserts
β	remaining back thickness
φ	insert diameter

Abbreviations

AD	Algorithmic Design
BIM	Building Information Modeling
CAD	Computer-Aided Design
DP	Design Procedure
DFM	Design For Manufacturability
PDP	Parametric Design Procedures
PD	Parametric Design
VPL	Visual Programming Language

Table of contents

Abstract

Acknowledgements

Symbols and abbreviations

Figures

Tables

1 Introduction

Companies across a range of industries now prioritize 3D modeling development. Increasing the market and boosting the company's future profitability helps the engineering teams and the business sector. Thanks to high-performance and sophisticated software that is readily available everywhere, many research and development initiatives can now improve this trend. Using 3D development tools can make these tasks a lot easier.

The mission of the Finnish multinational Metso Corporation is to support companies involved in the mining, aggregate, and recycling processes by offering them equipment, services, and technology. Crusher wear teams in Metso company work on the chamber optimization and development of the product in a way that meets the specific customer requirements. Some products that have a complicated design process are designed by the Metso crusher wear part development team for different kinds of crushers like primary gyratory, and cone crushers with various sizes according to the customer needs in the mining and aggregate industry. Crusher wear parts are shown in Figure 1 which is the mantle, and bowl liner from right to left, and Figure 2 which is Polycer protective wear. Those products have been developing for many years in Metso crusher wear part teams that have an insert on the crushing surface for increasing the time life of the product and efficiency during operation. Those inserts have specific features in length and diameter.

Figure 1. Mantle and bowl liner for HP cone crusher machine

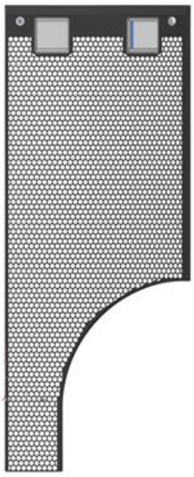


Figure 2. One liner from crusher protective wear part (Polycer)

Complex patterning elements were introduced into the crusher wear product design, which required a lot of work to produce and had some challenges when making alterations later. The design method for the crusher is usually done by manually using the NX 3D model, assembly environment, and different types of sketching in plans, revolve, patterning feature, and subtraction. Sometimes expressions are utilized for the insert location which is useful in reducing the time of designing.

The goal of the thesis is to speed up and improve the design process by utilizing parametric methods in 3D model design. This is becoming a reality with the aid of generative design in 3D model software. Since most product design is done manually, generative design could be a useful technique for this research, especially when concentrating on the algorithmic features. The 3D model case that is going to be investigated in this research has a similar design feature that Metso crusher wear parts have as shown in Figure 3.

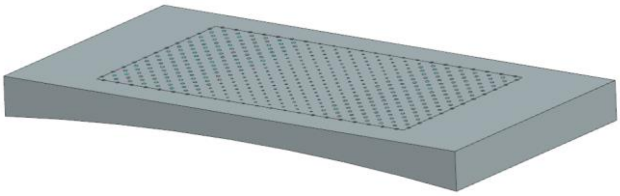


Figure 3. 3D model sample

In this thesis, the capability of the algorithmic feature for 3D modeling development of the complicated patterning feature for the mentioned 3D model case will be investigated.

1.1 Research Background

Modern engineering design heavily relies on Computer-Aided Design (CAD) systems, which have progressed from simple 2D sketching tools to intricate 3D design suites. These advanced computer-aided design (CAD) systems provide features for teamwork, intricate modeling, simulation, and copious documentation. (Stroud and Nagy 2011, p.53.)

Using computer tools to assist in the design process can be possible by using Computer-aided design, or CAD. It helps designers and engineers to create 2D and 3D models to enhance workflow, productivity, and design quality. Additionally, it routinely updates a database of manufacturing designs. CAD software is used in generative design to research and generate design options depending on inputs including design objectives, efficiency, structural requirements, materials, production methods, and cost constraints. (Rawat and Tiwari 2022, pp.275-276.)

Generative design in 3D model software can be achieved through parametric modeling, and algorithmic features facilitate the application of generative modeling in design. Unlike traditional 2D or 3D sketching, parametric modeling entails the creation of mathematical models of objects with parameters that allow for effective product alteration and reconstruction as well as the prevention of fundamental modeling errors. (Kravchik 2023, p.2.)

Parametric Design Procedures (PDPs) offer a technique that combines Design Procedures (DP) and Parametric Design (PD) to address the problems of the conceptual design stage. PDPs use a mathematical technique and parameters as inputs to generate and explore design possibilities dynamically. PDPs offer a wide range of designs compared to the DP which has limitations by design tree. This feature allows for several formal investigations inside a single model. This feature of the method is because of goes beyond the capabilities of parametric modeling and enables the 2D and 3D modification of objects, including non-closed shapes in the conceptual design phase, a designer may alternate between extrusion and rotation to produce various iterations of a model. PDPs employ a historical-based framework, which allows designers to alter any step at any point in time. (Abdullah and Kamara 2013, p.337.)

To ensure that models update automatically as dimensions change, parametric computer-aided design (CAD) relates constraints to dimensions. Adjustments must be made manually no more because of this automation. Product family design benefits greatly from its application. Teams can update models more effectively and comprehend how changes affect them by using parameterization. Easy construction of entire product families is the key advantage. Faster production and integration of the manufacturing process are also made possible. To link constraints correctly without creating interference or accuracy problems, more time is needed upfront. For significant automation-related downstream productivity benefits, more development work is thus required as a trade-off (Kyratsis and Manavis and Davim 2023, p.207.)

Today's generative design technology makes programming tools more affordable for designers and makes it much more accessible. This accessibility allows various ideas to be built efficiently in a shorter amount of time than with previous approaches. Although it was initially utilized in architecture, generative design is currently used in other industries, such as industrial and jewelry design. During the initial design process, it is particularly helpful for creating multiple design options quickly. For instance, many jewelry concepts can be produced more quickly and easily utilizing generative technologies. This ensures manufacturability by interacting with DFM software. Comparing generative design to manual sketching/2D increases designer productivity and creativity, which helps with early ideation and decision-making. Because much initial design exploration is handled computationally rather than by hand, it saves time. Therefore, before choosing final designs, engineers can choose from quick concept creation along with production input (Jaisawal and Agrawal 2021, p.2). As a result, manufacturing has experienced a change in basic assumptions, with generative design enabling the quicker production of distinctive and customizable products (McKnight 2017, p.180.)

Generative design in this thesis is going to be illustrated by using no code, non-code which is visual programming, that does not need coding. In visual programming, a no-code programming technique, code is encrypted into components. Every one of these components has inputs and outputs. Wires can be used to connect these components. Compared to conventional programming, visual programming is much more straightforward, making it easy for beginners to learn. (Ajouz 2021, p.185.)

1.2 Literature survey

Some of the articles in this section revolve around the general idea of the parametric design and how it can be helpful for the 3D development of the product. The rest of the articles firstly focus on generative design and how it works in favor of the parametric design then secondly the algorithmic design in 3D modeling software. There are not many articles in the current literature on generative design that address algorithmic features in any kind of 3D modeling software. To thoroughly explore and understand algorithmic features in different 3D model software, further research in this area is necessary, as this knowledge gap highlights.

In Li-li (2015), the design process is improved through the proposal to utilize knowledge engineering theory and methodologies for extracting and organizing expertise pertinent to the structure and design of any product. This knowledge may then be applied to create parameterized variation product templates with user-friendly interfaces using 3D model software. These templates make it easier to quickly create new product designs and promote the effective reuse of design knowledge. By simplifying the design process for every product, this approach aims to boost overall productivity in product development. Two different templates were defined in research, which was investigated by Li-li (2015), called assembly design templates and parts design templates. The template design in Li-li's (2015) research included: the mining of structural knowledge, which involves breaking down a pressure valve's construction to get the general driving structure parameters. Product structure knowledge mining, after decomposing the product structure into a tree, the article describes how to examine global design parameters and component interactions. Top-down and part-by-part size relationship analysis are both used to achieve this. NX software was utilized as a tool for designing design templates. This includes creating the template interface and establishing a hierarchical relationship structure. Make a Product Assembly Model, the assembly model is created from the bottom up in compliance with the structure. (Li-li 2015, p.290.)

In a study by Aranburu and Camba and Justel and Contero in 2023, they determined the optimal modeling interpretations of the parametric CAD explicit reference methodology. The explicit reference modeling approach described in the research results focuses on parametric CAD modeling with explicitly structured methods. The method includes the reconstruction of affected objects outside of groups, local variables such as fillets and chamfers, and integrated operations using Boolean functions. The course recognizes the challenge of measuring design concepts in CAD models and the effort required for manual model validation. They emphasize the importance of using local variable properties that are close to the geometric features of their variables to reduce dependencies between features by reducing parent-child relationships. Previous studies have proposed parametric modeling techniques such as explicit reference modeling for a consistent approach to flexible and reusable model design in the automotive industry. (Aranburu and Camba and Justel and Contero 2023, p.3.)

The impact of visual coding on product layout education, particularly in Industrial Design and 3D Design disciplines is highlighted by Novak and Loy (2017) in their research. It explores how visual programming languages (VPLs) are integrated into layout software, permitting designers to interact with coding visually and intuitively. The paper argues that VPLs blur the boundaries between layout, programming, and electrical engineering, permitting designers to control their designs in new approaches. It highlights the changing position of designers, emphasizing the importance of digital technology in layout schooling and the need for designers to conform to the virtual landscape. The record affords a case observation wherein Industrial Design college students favored visual coding over textual content-based coding for growing prototypes, indicating the capability of VPLs to decorate hassle-fixing and layout outcomes. Overall, it underscores the significance of VPLs in empowering designers to create more complicated merchandise by integrating electronics and parametric systems into their designs. (Novak and Loy 2017, pp. 237-238.)

Rawat and Tiwari (2022) compared Algorithmic features in NX and Grasshopper’s program for modeling a conical spiral. Siemens NX and Grasshopper both use geometric algorithms to computationally produce designs; however, research comparing their generative design capabilities discovered that Grasshopper has more functionality and is now used more often in architecture. Though it is still a relatively new feature, Siemens NX's inbuilt algorithm feature has shown promise as a generative technique that allows complex forms and associative, automatic iterative modeling without the requirement for third-party plugins. While Grasshopper leads in modern use and tools for generative design in architecture, Siemens NX's Algorithmic Feature, with more work, provides a built-in, easily accessible place to begin investigating integrated generative engineering without the need for further software or licensing. Designers can generate a multitude of design alternatives for conical spiral using an iterative approach with preset constraints and inputs. Designers assess and refine designs at several iterations to ensure they align with standards and requirements, as illustrated in Figure 4. Visual programming generative tools are part of generative design in both software that they were using for their research and are increasingly common in industry and research because those tools make geometric algorithms, which are essential to the design generation process, visually expressed. (Rawat et al. 2022, p. 281.)

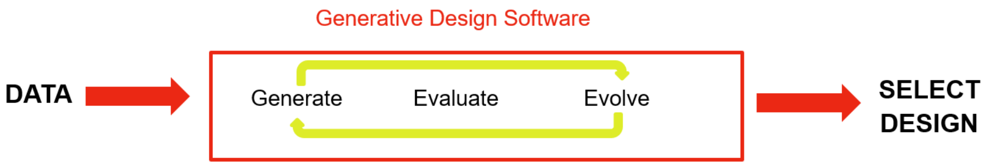


Figure 4. Design evolution with generative design software (Rawat et al. 2022, p. 276).

In a study by Castelo-Branco and Caetano and Leitão in 2021, they investigated some case studies, one of which was the German Formstelle pavilion, and compared how geometric limits might be represented and worked within the paradigms of computer-aided design (CAD), building information modeling (BIM), and algorithmic design (AD). The pavilion's front design, with its hexagonal holes creating a rippling visual effect, serves as a case study. The model is hard to design in CAD and BIM, requiring time-consuming updates and human modifications. However, AD allows for the explicit specification of design criteria, which improves the efficacy of testing variations. The algorithmic approach involves resizing hexagonal pieces in response to attractor curves and mapping them onto a surface grid as shown in Figure 5. (Castelo-Branco and Caetano and Leitao 2022, p.533.)

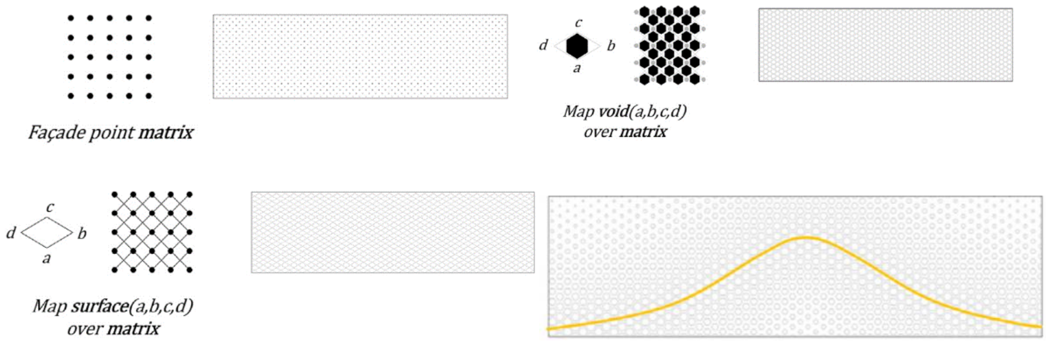


Figure 5. The Formstelle building’s algorithmic model is implemented step-by-step (Castelo-Branco et al, 2022, p.533).

They discovered that, in comparison to CAD and BIM, Algorithmic Design is more effective at expressing and comprehending designs. It understandably demonstrates design concepts, effectively accomplishes design goals, and prevents misconceptions. According to the study, by combining visualization with algorithmic design, designers can investigate possibilities and observe the effects of making changes. (Castelo-Branco et al., 2022, p.534.)

The parametric design of a truss structure that can switch between an N-truss and a Vierendeel truss is described in Ajouz’s article. First, a top-down and bottom-up node grid is built. These nodes are connected by lines to create the original Vierendeel shape. Diagonals are carefully inserted in two directions to turn this into an N-truss: from top left to bottom right, and vice versa. The start and finish positions of the diagonals are determined parametrically using formulas that consider the number of segments; this allows for variations in height and length. The result is an adaptable N-truss design which is shown in Figure 6. It is determined by an additional parameter whether the geometry is an N-truss or a Vierendeel. All things considered; this is an adjustable method that produces effective truss constructions. (Ajouz 2021, pp.188-190.)

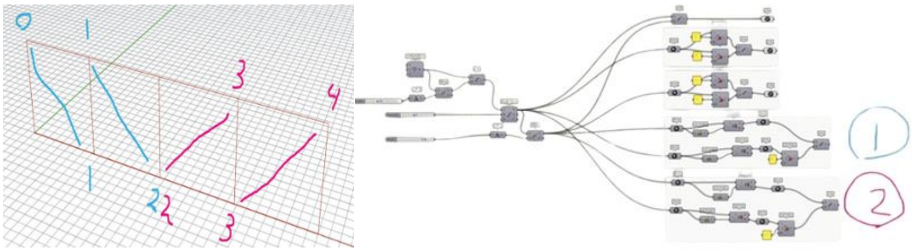


Figure 6. Grasshopper and computational plan for designing N-truss (Ajouz 2021, p.190).

The GDNet generative design framework is presented by Jang et al. to optimize design variety in reinforcement learning. The resource-intensive topology optimization on CPUs is replaced with significant human guidance by using the TopOpNet neural network on GPUs to approximate optimization. This method creatively uses reinforcement learning to make the generative design process substantially more automated, resulting in a range of designs produced quickly from a reference image. (Jang and Yoo and Kang 2022, p.5 & p.16.)

1.3 Research Problem

Previous researches that have been reviewed recently mostly focus on the algorithmic features of Grasshopper software and emphasize the architectural aspect. In terms of industrial engineering design and mechanical engineering, different types of academic sources do not have much information about the algorithmic feature usage in 3D model software and there is a lack of details about the functionality of algorithmic feature for 3D modeling development of products. Numerous problems that could be categorized as research challenges were found in the survey and literature evaluation. The following is a list of the recognized research problems.

1. Features for developing the modeling complex patterned components cannot easily be utilized in 3D model software and there are different ways to reach patterning features in different kinds of 3D modeling software.
2. There are lots of parameters that affect the designing concept, and mathematical knowledge and visual coding skills are needed for applying those with algorithmic features in 3D model software.

3. Converting the knowledge and mathematical rules in designing patterning feature items to algorithmic features needs specific rules that are not easy according to the patterning requirement for design.
4. Applying parametric design for different geometry and 3D sample cases that have complex patterning features needs to be altered quickly according to the design requirement and it is not easy to make any modeling changes in designing.

1.4 Research Questions

In reviewing various articles and examining a new product's innovative approach, along with the lack of information regarding algorithmic features and from the research problems list, several questions have emerged.

1. What is the general approach to developing complex patterns in engineering design when such design patterns are not easily accessible in standard methods and software libraries?

2. How it is possible to make the patterning feature in the algorithmic feature?

3. What would be the best method to do that kind of patterning and expand it to the different kinds of products?

4. There are numerous effective design parameters that designers must consider when creating 3D models with complex patterned components. How can these parameters be integrated into algorithmic features within 3D modeling software?

5. How can algorithmic features be modified as quickly as possible without necessitating significant alterations to the main structure?

1.5 Research Scope

The scope of the thesis is to the development of 3D modeling of the crusher wear parts by generative design tools focusing on NX software which is one of the Metso company software designing tools. Going through the complex pattern for the inserts in a 3D model case is going to be considered. The ability to utilize algorithmic features for different types of 3D model geometry is considered. It is important that each improvement through the node in the algorithmic feature node library can be an asset for designs. Meanwhile, some samples are going to be mentioned which are highly related to the target model.

1.6 Structure of Thesis

The first chapter of the thesis is about why this research is important to designers and how it can be useful to save time and increase productivity. Chapter 2 describes the methodology used in the study, focusing specifically on the use of algorithmic feature nodes for 3D modeling. Chapter 3 shows the results of applying the parametric design approach and using the algorithmic feature in different geometry. The advantages of the methods are also discussed. In the final chapter, the new computer-aided design method is compared with traditional handmade methods. The pros and cons of each and suggested hypotheses for future research are investigated.

2 Algorithmic feature method in 3D modeling

Parametric design is a popular tool in industrial design that is used to quickly build a series of items and create detailed product models, particularly for complicated surfaces. Creating distinct types within the same series of 3D models does not require designers to start from scratch to design the model. They can expand the designing method upon the base algorithm that already exists, streamlining the design process, cutting down on time, and eliminating repetitious effort. Furniture, appliances, and other household goods can be designed with the help of parametric design (Sun, et al. 2019, p.3). The methodology is based on parametric design and focuses on finding the best solutions using 3D model software tools. The research intends to investigate the many functions of 3D model software in enabling user expression and equation formulation for parametric design. The main goal is to identify efficient parametric design techniques in the 3D program and assess their appropriateness for these kinds of uses. The research approach will be covered in detail in this chapter, with an emphasis on the parametric design elements required to answer the research questions that have been found. An overview of general information and details of algorithmic modeling in NX software will be given in the first section of this chapter. Several examples illustrating techniques useful for this study will next be carefully analyzed and presented. Finally, the general idea of creating intricate designs in Metso's case will be presented to wrap up the chapter.

2.1 Generative and parametric design

The process of designing for the next generation of customers is evolving due to a new technology called generative design. It improves design possibilities outside of conventional design techniques using algorithms. To identify the best solution, generative design quickly examines hundreds of potential configurations, imitating the natural evolutionary process. A portion of the procedure involves introducing design objective criteria including material qualities, load conditions, and restrictions. Engineers and designers work together with computers to complete the final design, and via experimentation and system learning, each iteration yields the best answer. Generative design assists major manufacturers in reducing the number of components, increasing efficiency, speeding up manufacturing processes, and creating unified, individualized goods. It has been applied to many different industries, such as aircraft, to build compact systems that require few resources and save fuel. Because it creates new avenues for creativity and efficiency, generative design is revolutionary in the fields of technology and design (McKnight, 2017, pp.177-180). An invaluable additional resource for engineers is visual programming. Parametric design offers a wide range of opportunities for optimizing design project solutions with its abundance of available plugins. Improving the capacity to deconstruct intricate structures into a coherent series of actions is the main goal of training, which is necessary to realize its full potential. When using parametric design to its full potential in the search for better solutions, this ability is very important. (Ajouz, 2021, p.194.)

2.2 Algorithmic feature

Algorithm features are typically found in software or programs that employ algorithms to carry out operations. There are several applications for algorithmic features in this scenario. Designers can build and work with geometric shapes using parametric modeling tools and mathematical parameters (Cucoş et al., 2018, p2). Customized designs that adhere to preset requirements and limitations can be created using generative design approaches. This may lead to product parts that are structurally sound and more effective in terms of performance and material handling. Complex patterns or models can be generated routinely, instead of by hand, using modeling techniques. Iterative design enhancements can be accomplished using optimization techniques to achieve performance goals like increasing structural integrity or lowering material consumption. By adding algorithmic elements to 3D modeling approaches for product parts, engineers may increase the precision, effectiveness, and longevity of these crucial components improving productivity and dependability in the manufacturing and recycling industries. (Li & Lachmayer, 2018, p.2.)

This algorithmic feature contains some parts in which the designer can create several nodes in the logic editor. Node can be a feature operation (for creation or modification of geometry) a mathematical/logical operation (the actual “rules” reside here or simply a selection operation for selecting existing geometry.

2.2.1 Algorithmic feature user interface

The algorithmic feature's user interface is designed with ease of use in mind, with important functions like Snippet, Markup, Group node, Update Control, Rule Management, and Rule Navigation at the top. The left side is devoted to designers creating nodes and rules and includes a Search Pane, Node Pane, and Rule Input Pane that are going to be explained further in 2.2.3. There are two components on the main screen: a Part Window and the Logic Editor Canvas. Whereas the latter dynamically presents the relevant 3D model, the former lets designers apply Logic rules. The designer can alter the outputs and see real-time updates in this display, making it responsive and dynamic. Figure 7 shows the name of each option and section.

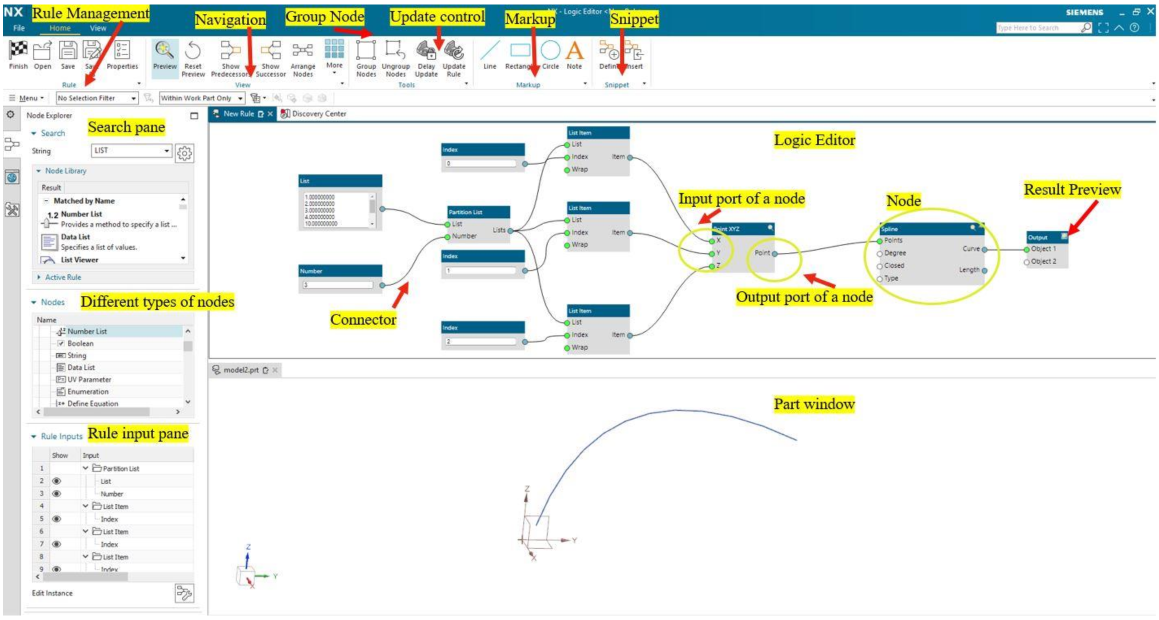


Figure 7. Algorithmic feature user interface

2.2.2 Nodes and Logic Editor

Every node has an input port and an output port with a preview toggle on each. By connecting one node's output to another's input, rules are created. Lists of objects that match the anticipated type or individual objects can be used as inputs and outputs. By choosing nodes and logic to produce output, users build algorithms in the Logic Editor. There is a preview button (Result Preview) on nodes that have topological or geometric output. Users can view the generated model preview by clicking this button, which opens a second model preview window (Part window) as shown in the 2.2.1 section. (Rawat et al. 2022, p.280.)

2.2.3 Rule input

The rule inputs in the algorithmic feature view correspond to each node that is created in the logic editor. This allows users to modify how input items are presented directly in the design environment of the 3D model. As a result, designers may easily adjust input parameters in the design environment, doing away with the necessity to do so in the algorithmic feature area. The Logic Editor automatically generates an accompanying dialog box when a designer creates a rule. Designers can examine the resulting dialog box by clicking the Edit Instance button in the Rule Inputs section of the Node Explorer. Designers can use drag-and-drop to change the order and grouping of the dialog box fields in the table included in the Rule Inputs section. Designers can also choose to hide dialog box fields that are not meant to be changed in everyday use.

2.3 Various types of lists in algorithmic feature

For algorithmic design procedures to effectively handle and organize data, various list operations are required. These lists serve as containers for various data objects such as points, bodies, faces, lines, length of lines, etc. Designers can manipulate these lists to choose, filter, and refine the data according to the design goal by utilizing mathematical and logical operations. For instance, lists make it possible for designers to quickly locate relevant data points, retrieve dimensions or locations, and carry out computations using the information they contain.

2.3.1 Demote List

When using algorithmic features, different kinds of lists might have an impact on the design process. In this section, one specific kind of list that is useful for designing is explored along with a thorough example. The two boxes (highlighted in yellow) in Figure 8 each have points on their surfaces with spatial coordinates as shown in Figure 9.

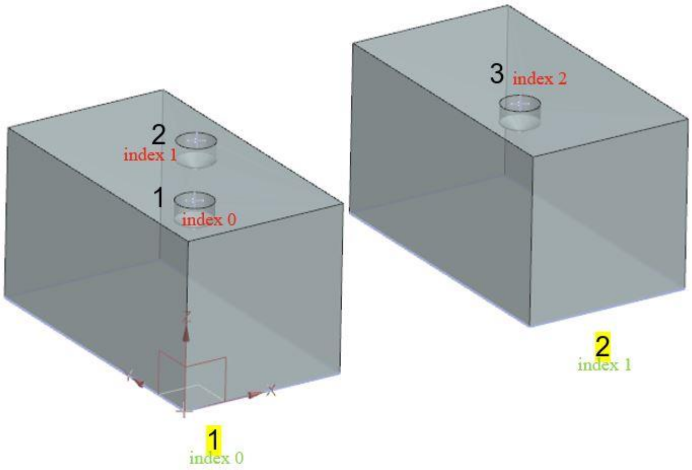
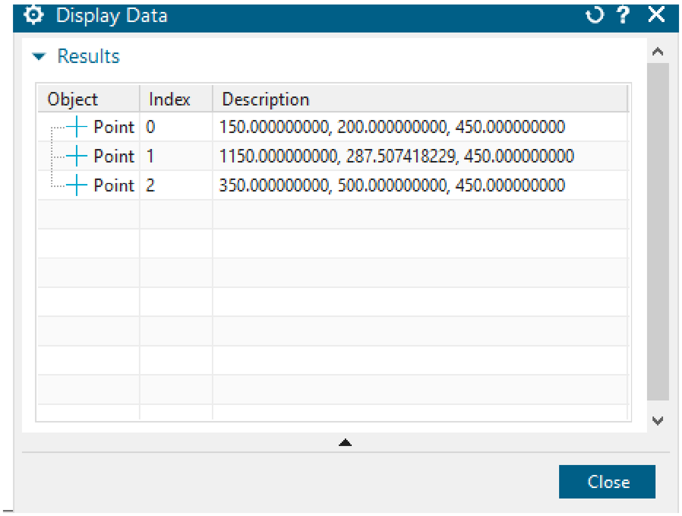


Figure 8. Two boxes with hole feature



Object	Index	Description
Point	0	150.000000000, 200.000000000, 450.000000000
Point	1	1150.000000000, 287.507418229, 450.000000000
Point	2	350.000000000, 500.000000000, 450.000000000

Figure 9. Spatial coordinates for hole's points in 2 boxes

The simple hole is utilized to create holes on the surfaces of two boxes. The positions and targets are chosen as points, with point indexes 0,1 and 2 corresponding to points 1,2, and 3 respectively on the body surface, as shown in Figure 8. The direction is specified as I=0 J=0, and K= -1. Also, for the target bodies, bodies one and two corresponding to the indexes 0 and 1 are selected. When sequences are selected in the way outlined in the previous description, wrapping behavior occurs. The reason for this is that point two (index 1) is on Box 1 (index 0) and it is not on the surface of Body 2 (index 1), a problem occurs as the algorithm moves from point one to point two and finally to Body 2. As a result, as the image 10 illustrates, this produces erroneous findings.

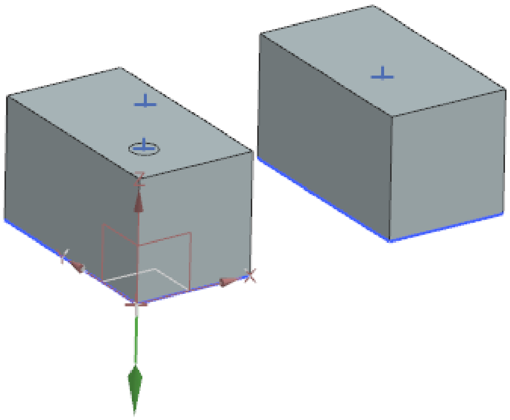
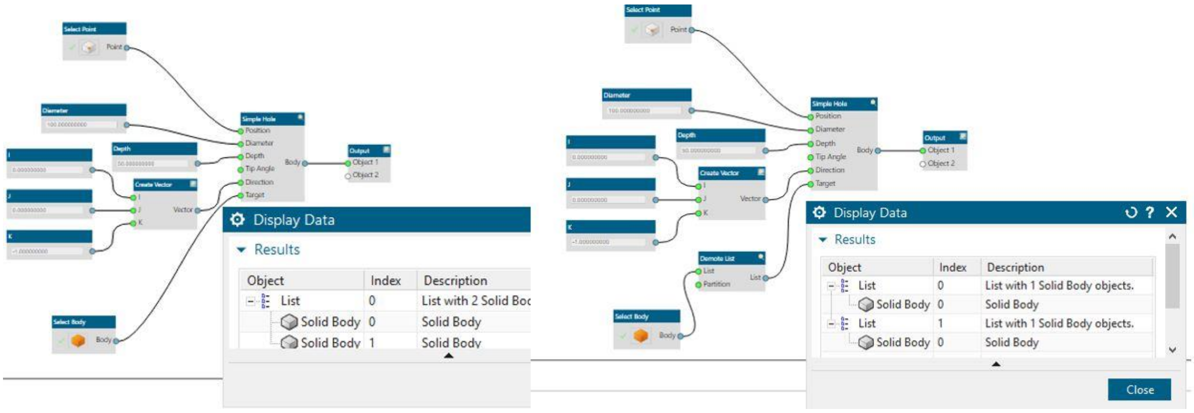


Figure 10. Wrapping issue and uncompleted model

Typically, designers want to select many target bodies at different points in any given sequence. It is anticipated that NX (probably a design program) would intelligently produce every possible hole on bodies while eliminating areas that are not practicable. To do this, the Demote list facilitates the use of a layered list of target bodies. Figure 11 illustrates two approaches to utilizing nodes. In the left option, without demoting the list, both bodies are placed in a single list at index 0. Here, Body Number 1 has an index of 0 in List 0, and Body Number 2 has an index of 1 in List 0. Conversely, the right option adopts demoting the list, ensuring that each body is assigned the index of 0 in separate lists. This use of demoting lists is a reasonable strategy to achieve correct and unrestricted design, effectively avoiding limitations in the design process.



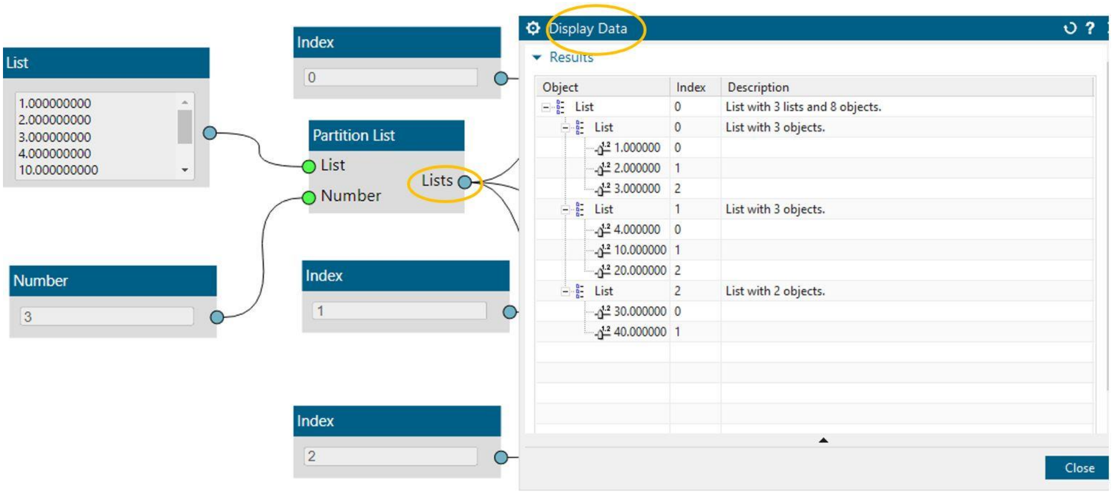
Object	Index	Description
List	0	List with 2 Solid Boc
Solid Body	0	Solid Body
Solid Body	1	Solid Body

Object	Index	Description
List	0	List with 1 Solid Body objects.
Solid Body	0	Solid Body
List	1	List with 1 Solid Body objects.
Solid Body	0	Solid Body

Figure 11. Demote list behavior

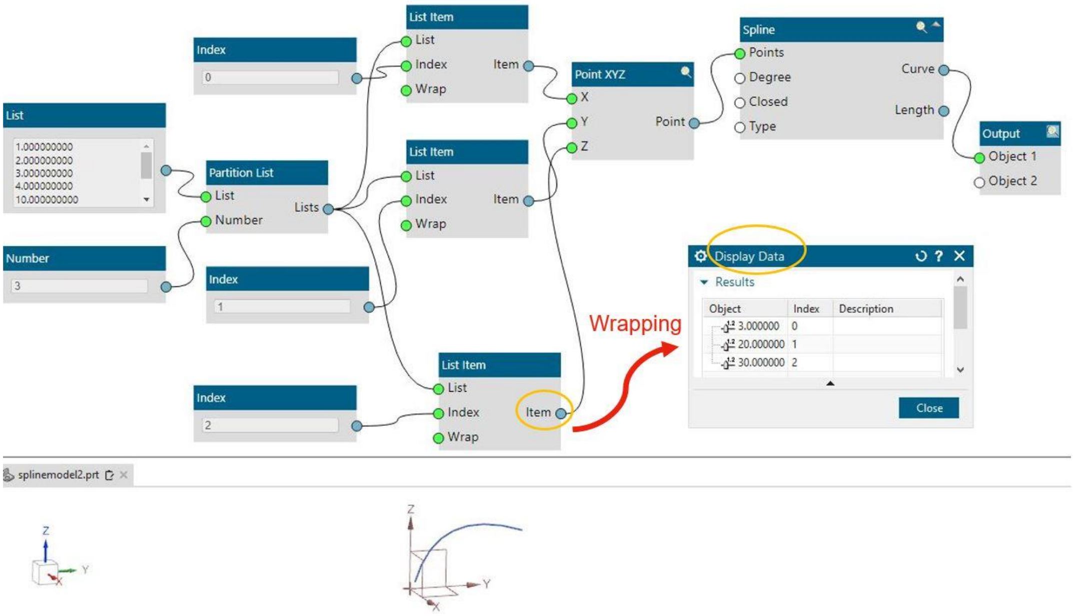
2.3.2 Partition List and List item

The partition list function can be utilized by designers to divide the flat number list of the algorithmic feature according to numerical criteria. Additionally, designers can accurately identify and extract the desired list for geometric or feature-related criteria by using the List item node and index as input. For example, choosing three points in space is the first stage in building a spline for a 3D model drawing. The stated points are broken down into sets of three in Figure 12 with each set denoting a set of points needed for the spline. Then, every point in a set has an index given to it. After that, the list items are selected to guarantee that every point in the matrix with the same index is grouped with every other point. The software displays its usual wrapping behavior when there is no information for a given index in the list and it automatically chooses index 0 instead of index 2 which is not available in list 2. This can be seen in Figure 13. Wrapping behavior is shown for the list item corresponding to the Z-point in Figure 13. This happens because List 2 in Figure 12 does not have index 2. As a result, the program chooses 30 by default, which is List 2's index zero, and causes the wrapping phenomenon that has been seen.



Object	Index	Description
List	0	List with 3 lists and 8 objects.
List	0	List with 3 objects.
1.000000	0	
2.000000	1	
3.000000	2	
List	1	List with 3 objects.
4.000000	0	
10.000000	1	
20.000000	2	
List	2	List with 2 objects.
30.000000	0	
40.000000	1	

Figure 12. Partition list operation



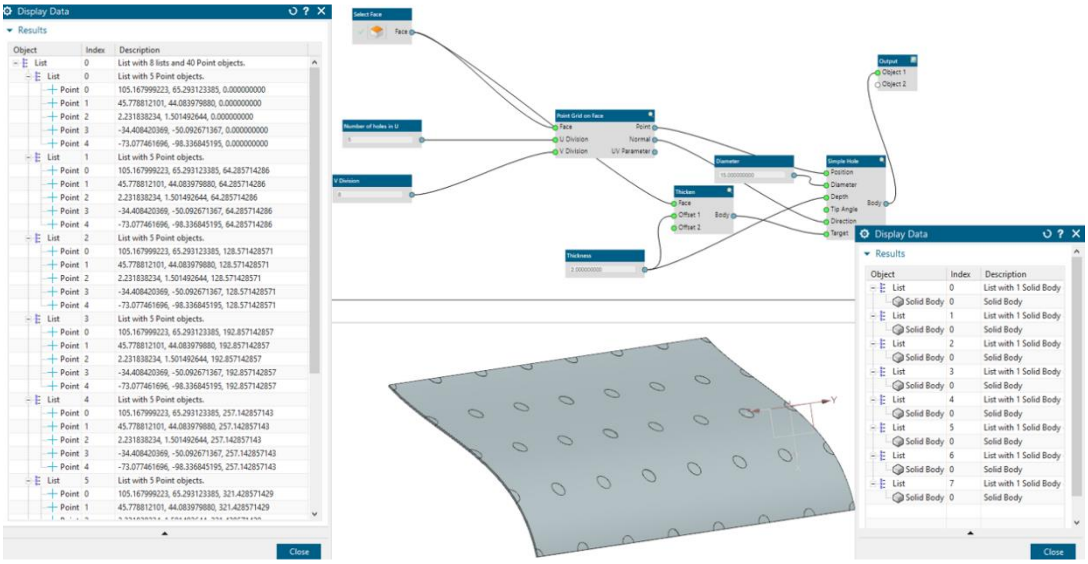
Object	Index	Description
3.000000	0	
20.000000	1	
30.000000	2	

Figure 13. Spline algorithmic feature

When it comes to incorporating different logical structures into a designer's workflow, the List Item node is a useful tool. As the partitioning and spline-building example above illustrates, it makes it easier to extract an index from a list.

2.3.3 Flatten List

A multi-level list may occasionally produce unexpected results that deviate from the planned layout. It becomes necessary to flatten the list to mitigate such difficulties. The example that follows shows why the list must be flattened to fix an issue with the model. The orientation of the basic holes is in line with the typical output from the Point grid on the face box, and the points are generated in eight lists (0-1-2-3-4-5-6-7). As a result, the target body is still one list, but the direction and placements of the basic points are arranged into eight lists of five points each. The issue stems from the preceding circumstance, which leads to the wrong choice of one body for eight lists. This has a detrimental influence on the result, hindering the appearance of holes on the surface and creating the 7 lists of solid bodies that are the same for one product and should be merged into just one body as shown in Figure 14. To do so, the flattened list should be used after getting the output of points and normal and connecting those to the flattened list to be able to make those in one list like in Figure 15.

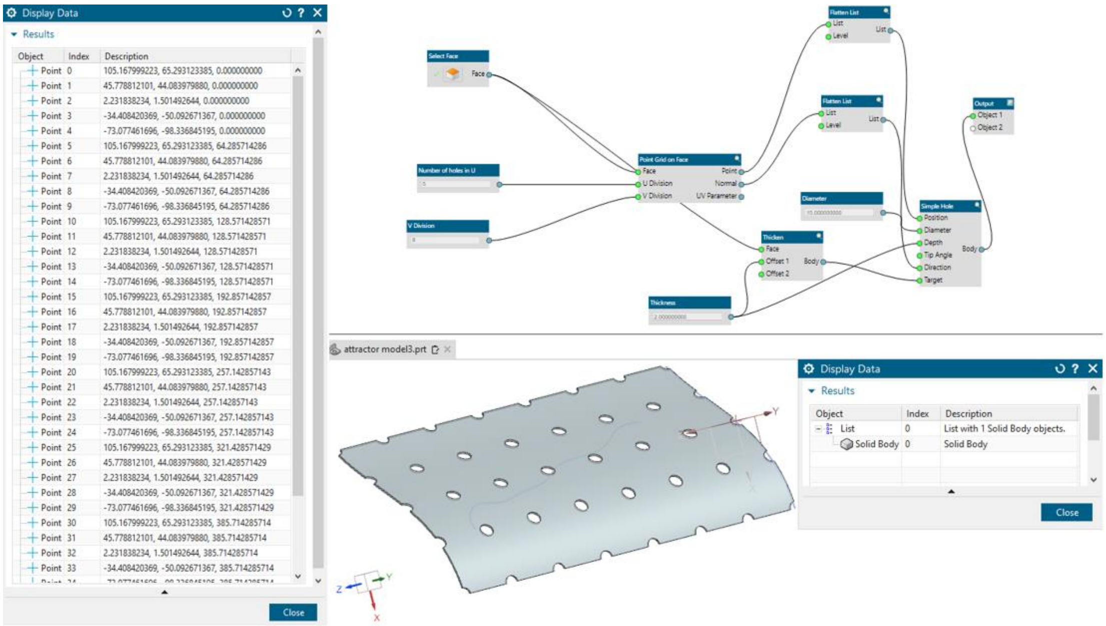


```
Object       Index  Description
List         0      List with 8 lists and 40 Point objects.
  List       0      List with 5 Point objects.
    Point    0      105.167999223, 65.293123385, 0.000000000
    Point    1      45.778812101, 44.083979880, 0.000000000
    Point    2      2.231838234, 1.501492644, 0.000000000
    Point    3      -34.408420369, -50.092671367, 0.000000000
    Point    4      -73.077461696, -98.336845195, 0.000000000
  List       1      List with 5 Point objects.
    Point    0      105.167999223, 65.293123385, 64.285714286
    Point    1      45.778812101, 44.083979880, 64.285714286
    Point    2      2.231838234, 1.501492644, 64.285714286
    Point    3      -34.408420369, -50.092671367, 64.285714286
    Point    4      -73.077461696, -98.336845195, 64.285714286
  List       2      List with 5 Point objects.
    Point    0      105.167999223, 65.293123385, 128.571428571
    Point    1      45.778812101, 44.083979880, 128.571428571
    Point    2      2.231838234, 1.501492644, 128.571428571
    Point    3      -34.408420369, -50.092671367, 128.571428571
    Point    4      -73.077461696, -98.336845195, 128.571428571
  List       3      List with 5 Point objects.
    Point    0      105.167999223, 65.293123385, 192.857142857
    Point    1      45.778812101, 44.083979880, 192.857142857
    Point    2      2.231838234, 1.501492644, 192.857142857
    Point    3      -34.408420369, -50.092671367, 192.857142857
    Point    4      -73.077461696, -98.336845195, 192.857142857
  List       4      List with 5 Point objects.
    Point    0      105.167999223, 65.293123385, 257.142857143
    Point    1      45.778812101, 44.083979880, 257.142857143
    Point    2      2.231838234, 1.501492644, 257.142857143
    Point    3      -34.408420369, -50.092671367, 257.142857143
    Point    4      -73.077461696, -98.336845195, 257.142857143
  List       5      List with 5 Point objects.
    Point    0      105.167999223, 65.293123385, 321.428571429
    Point    1      45.778812101, 44.083979880, 321.428571429
```

```
Object          Index  Description
List            0      List with 1 Solid Body
  Solid Body    0      Solid Body
List            1      List with 1 Solid Body
  Solid Body    0      Solid Body
List            2      List with 1 Solid Body
  Solid Body    0      Solid Body
List            3      List with 1 Solid Body
  Solid Body    0      Solid Body
List            4      List with 1 Solid Body
  Solid Body    0      Solid Body
List            5      List with 1 Solid Body
  Solid Body    0      Solid Body
List            6      List with 1 Solid Body
  Solid Body    0      Solid Body
List            7      List with 1 Solid Body
  Solid Body    0      Solid Body
```

Figure 14. Model without Flatten list node

In Figure 15, the position data and final output are displayed. It is crucial to flatten the input lists of points and directions for the simple hole input node to solve this problem. To attain the intended result, they can be combined into a single flat list of points, a single flat list of directions, and a single target input body. Figure 15 shows the revised of the previous logic that was created without using the Flatten list. The list after using flatten shows the list of the points which all are in the one list instead of the 8 different lists.



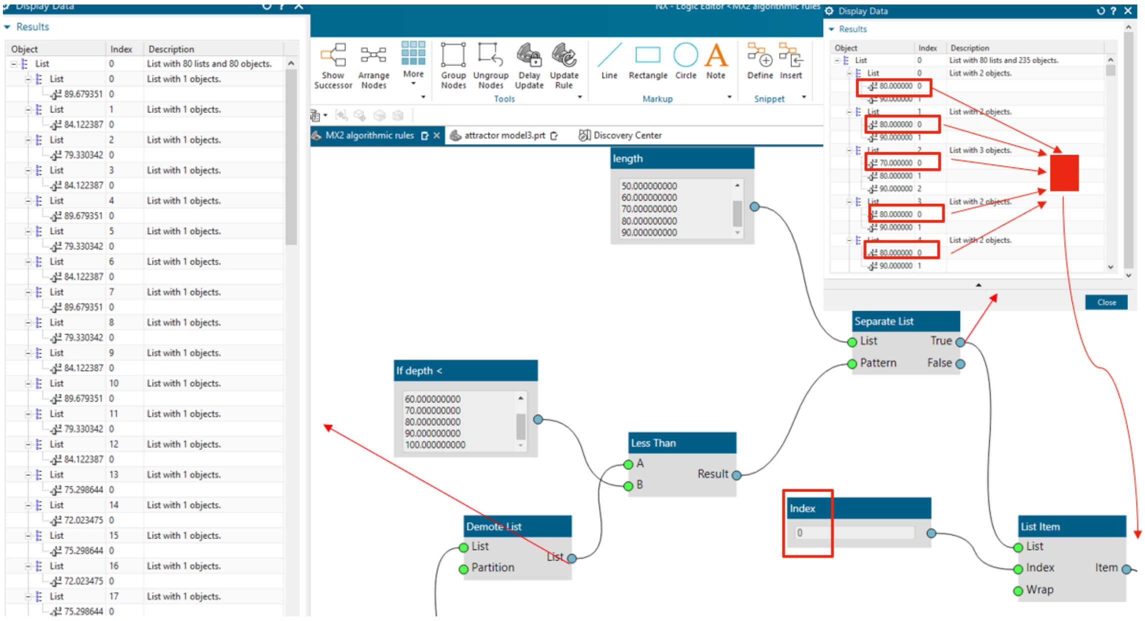
Object	Index	Description
Point	0	105.167999223, 65.293123385, 0.000000000
Point	1	45.778812101, 44.083979880, 0.000000000
Point	2	2.231838234, 1.501492644, 0.000000000
Point	3	-34.408420369, -50.092671367, 0.000000000
Point	4	-73.077461696, -98.336845195, 0.000000000
Point	5	105.167999223, 65.293123385, 64.285714286
Point	6	45.778812101, 44.083979880, 64.285714286
Point	7	2.231838234, 1.501492644, 64.285714286
Point	8	-34.408420369, -50.092671367, 64.285714286
Point	9	-73.077461696, -98.336845195, 64.285714286
Point	10	105.167999223, 65.293123385, 128.571428571
Point	11	45.778812101, 44.083979880, 128.571428571
Point	12	2.231838234, 1.501492644, 128.571428571
Point	13	-34.408420369, -50.092671367, 128.571428571
Point	14	-73.077461696, -98.336845195, 128.571428571
Point	15	105.167999223, 65.293123385, 192.857142857
Point	16	45.778812101, 44.083979880, 192.857142857
Point	17	2.231838234, 1.501492644, 192.857142857
Point	18	-34.408420369, -50.092671367, 192.857142857
Point	19	-73.077461696, -98.336845195, 192.857142857
Point	20	105.167999223, 65.293123385, 257.142857143
Point	21	45.778812101, 44.083979880, 257.142857143
Point	22	2.231838234, 1.501492644, 257.142857143
Point	23	-34.408420369, -50.092671367, 257.142857143
Point	24	-73.077461696, -98.336845195, 257.142857143
Point	25	105.167999223, 65.293123385, 321.428571429
Point	26	45.778812101, 44.083979880, 321.428571429
Point	27	2.231838234, 1.501492644, 321.428571429
Point	28	-34.408420369, -50.092671367, 321.428571429
Point	29	-73.077461696, -98.336845195, 321.428571429
Point	30	105.167999223, 65.293123385, 385.714285714
Point	31	45.778812101, 44.083979880, 385.714285714
Point	32	2.231838234, 1.501492644, 385.714285714
Point	33	-34.408420369, -50.092671367, 385.714285714

Object	Index	Description
List	0	List with 1 Solid Body objects.
Solid Body	0	Solid Body

Figure 15. Flatten list node operation

2.3.4 Separate list

The designer can choose from a range of indexes a certain criterion by using a separate list. As a result, the precise values required to create a sketch and construct the geometry can be chosen by designers. It also enables designers to create models or rules that let them select criteria list-wise depending on input patterns. Figure 16 is an example of how to use this algorithmic feature list. In that example, getting true as an output from a separate list is following one rule from pattern input, and by implementing that rule in the input list of the separate list, the designer can get the desired output and then by using the list item can get the specific value that needs for design. For instance, in this case, index 0 is chosen to get the first value out of the true list. As illustrated in Figure 16, the output of the Demote list contains different lists with index 0, in list 0 for instance, the value is 89.67 which after that it goes through the next node to compare with different items. 89.67 is less than 90 and less than 100 but not less than 80, so the output from less than node for that list 0 index 0 would be 90 and 100. The output 90 and 100 is going to the input for the separate node which is Pattern, and it follows the values. According to the length list values, for the 90 and 100, the 80 and 90 would be the output from the separate list which is shown in list 0 and index 0 and index 1 in display data of true output in Figure 16.



Object	Index	Description
List	0	List with 80 lists and 80 objects.
List	0	List with 1 objects.
89.679351	0	
List	1	List with 1 objects.
84.122387	0	
List	2	List with 1 objects.
79.330342	0	
List	3	List with 1 objects.
84.122387	0	
List	4	List with 1 objects.
89.679351	0	
List	5	List with 1 objects.
79.330342	0	
List	6	List with 1 objects.
84.122387	0	
List	7	List with 1 objects.
89.679351	0	
List	8	List with 1 objects.
79.330342	0	
List	9	List with 1 objects.
84.122387	0	
List	10	List with 1 objects.
89.679351	0	
List	11	List with 1 objects.
79.330342	0	
List	12	List with 1 objects.
84.122387	0	
List	13	List with 1 objects.
75.298644	0	
List	14	List with 1 objects.
72.023475	0	
List	15	List with 1 objects.
75.298644	0	
List	16	List with 1 objects.
72.023475	0	
List	17	List with 1 objects.
75.298644	0	

Object	Index	Description
List	0	List with 80 lists and 235 objects.
List	0	List with 2 objects.
80.000000	0	
90.000000	1	
List	1	List with 2 objects.
80.000000	0	
90.000000	1	
List	2	List with 3 objects.
70.000000	0	
80.000000	1	
90.000000	2	
List	3	List with 2 objects.
80.000000	0	
90.000000	1	
List	4	List with 2 objects.
80.000000	0	
90.000000	1	

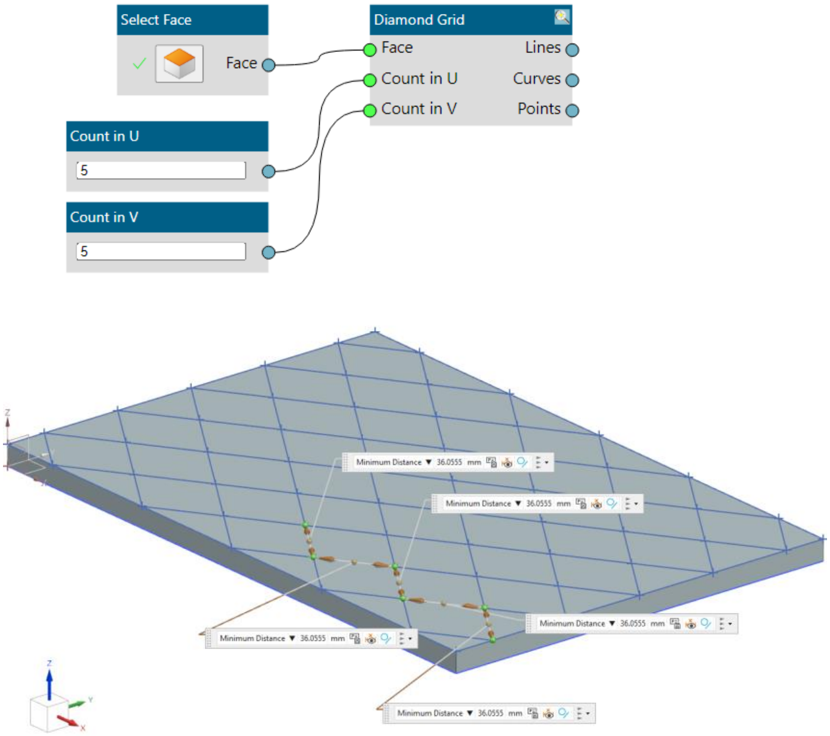
Figure 16. Separate list node operation

2.4 Point position by using a grid

By utilizing various techniques for positioning points on faces and surfaces, designers can utilize algorithms to generate distinctive features. They can improve their designs and get distinct outcomes by using different kinds of grids.

2.4.1 Diamond grid

This grid is defined by the U and V numbers for the number of points in 2 directions according to the geometry of the 3D model. For instance, in Figure 17 the surface of the 3D model is grided by that diamond grid. The 3D model with the size L=300 mm, W=200 mm, and t=10 mm can have a griding that is shown in Figure 17 and the distance between points can be the same 36 mm.



Count in U: 5

Count in V: 5

Figure 17. Griding on the surface using a diamond grid

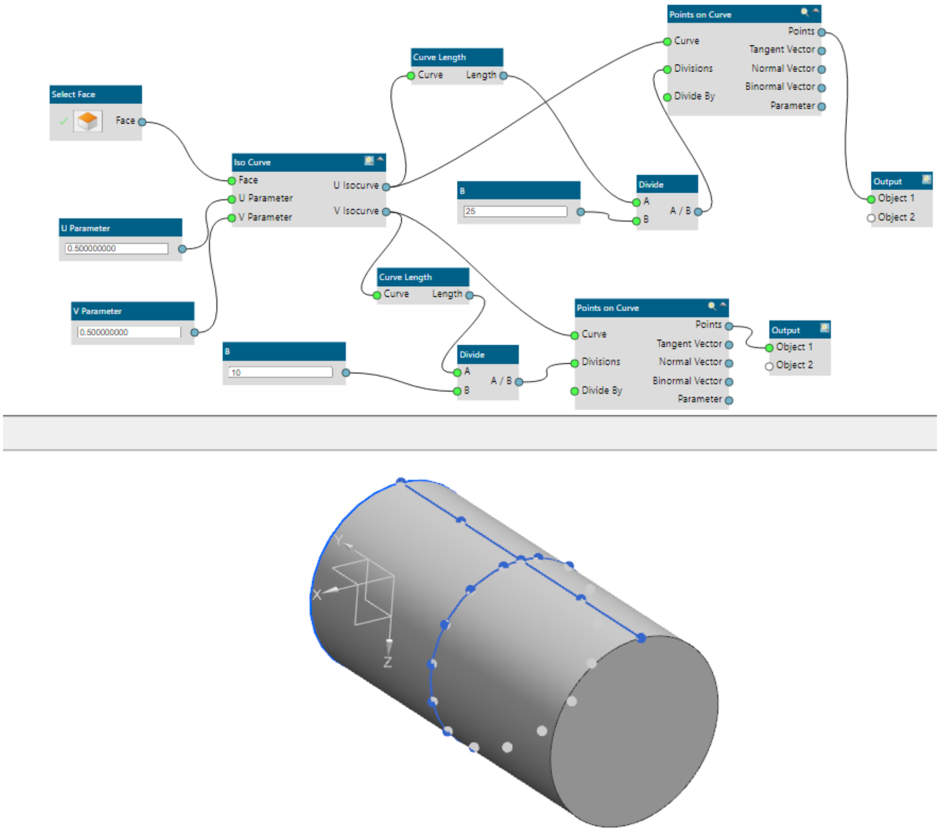
2.4.2 Iso curve and points on the curve

This feature is useful when a designer wants to locate a line on a surface and after that use that line for a specific purpose. For instance, using the points on curve nodes after creating a line on the surface. The iso curve location can be specified with the value less than 1 mm as it is shown in Figure 18, the curve length can be utilized for dimensioning the length of the curve in any location. In Figure 18, the status of the curve is the circumference of the circle, so the curve length should follow the formula below where the variables *C* represents the circumference, *r* represents the radius, and *d* represents the diameter of the circle which is created by the iso curve node on the outer surface of the cylinder is Figure 18.

$$C = 2\pi r \tag{1}$$

$$C = \pi d \tag{2}$$

By knowing about the circumference, the distance between the points can easily defined by dividing by the required value.


U Parameter 0.500000000

V Parameter 0.500000000

B 25

B 10

Figure 18. Griding on a surface using points on the Iso curve

2.5 Sample Case Study Requirement

The 3D model of this research has some features that Metso crusher wear parts have but with different geometry. The details geometry of the model can be seen in Figure 19.

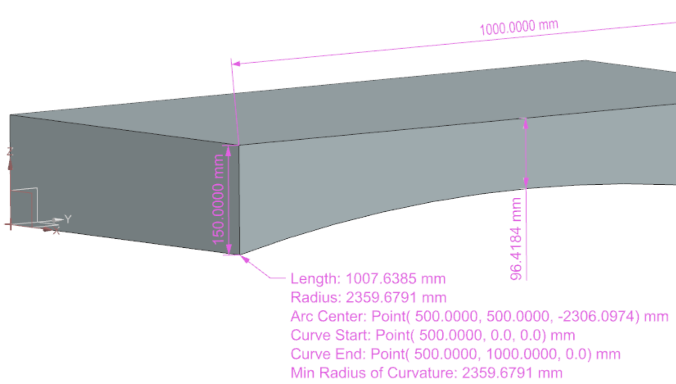


Figure 19. First model 1000×500×150

The bowl liner and mantle found in wear components for crushers are similar in design to this model. It is designed to demonstrate the idea and provide outcomes using both algorithmic and manual modeling. This new approach's advantages and timing are going to be evaluated later. Figure 20 depicts the visible inserts on the flat surface of the 3D model. Once these are inserted, the remaining back thickness (β), an important component of the modeling feature, must be considered by the designer to have the required limited value.

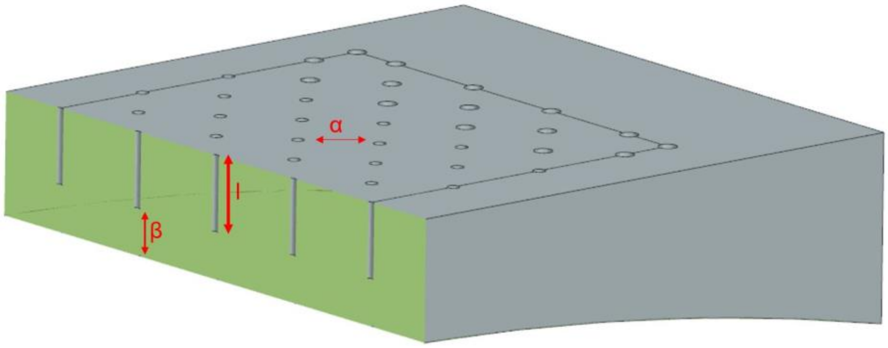


Figure 20. Inserts and design parameters in the First model 1000×500×150

The insert length and diameter are listed in the table1. Designers need to consider how altering insert diameters and lengths would affect the remaining back thickness while creating various models. Because each insert length (l) corresponds to a specific diameter (φ), it is vital to adjust the insert length based on the thickness of the back (β). Designers must notice that modifications to the diameter (φ) could potentially impact the spacing or remaining thickness between inserts (α).

Table 1. Inserts details

Insert Length l [mm]	Insert Diameter φ [mm]
30	5
60	10
70	10
80	15
90	15

So, in the next section of the research, the two different types of modeling are going to be analyzed in timing efficiency for two different 3D models.

3 Results and discussions

In this section, one 3D model with a wide range of insert density on its surfaces will be analyzed. Two different types of design have been used for that purpose in two models, one with the size of 1000×500×150 and the other is 2000×1000×150. One is the algorithmic feature for putting different kinds of inserts according to the thickness of the model and the insert density of the surface. However, the design method for the other one is not parametric and is done manually. The other key issue is that the minimum thickness after putting inserts in the model should be the same as one is 30 mm and the other model should be 20 mm. Table 2 shows the first model's detail that will be designed in 2 approaches (algorithmic and manual) and Table 3 corresponds to the case 2 design, which will be done in two ways, same as case 1.

Table 2. Designing details for case 1

Parameter	Description	Value	Unit
L	3D model length	1000	mm
r	Arc radius of 3d model	2360	mm
W	3D model Width	500	mm
t	3D model thickness	150	mm
α	Inserts spacing	50-55	mm
β	Back thickness	30	mm

Table 3. Designing details for case 2

Parameter	Description	Value	Unit
L	3D model length	2000	mm
r	Arc radius of 3d model	2360	mm
W	3D model Width	1000	mm
t	3D model thickness	150	mm
α	Inserts spacing	30-35	mm
β	Back thickness	20	mm

3.1 Grid on outer surface

Gridding for the model is done in two different ways. In algorithmic feature griding in this model is followed by the diamond grid and the other models are done by sketching and defining points and using user expression, pattern feature, and mirror feature. Inserts location is inside and on the edges of the rectangular space, which has a length of 680 mm × 370 mm width, for all designs. On the left side in Figure 21, the way of manual modeling is illustrated that the designer needs to position the 5 points, and after that cylindrical shape with a hole feature should be created and then the designer should check the minimum back thickness (30 mm) and distance between inserts which should be around 50 mm, and diameter should according to the Table 1. If all those features meet the requirements for designs, then the designer can go forward to create the other points and inserts in the surface by using the patterning and mirror feature. The right side of Figure 21 contains the diamond grid for the model by using U and V values.

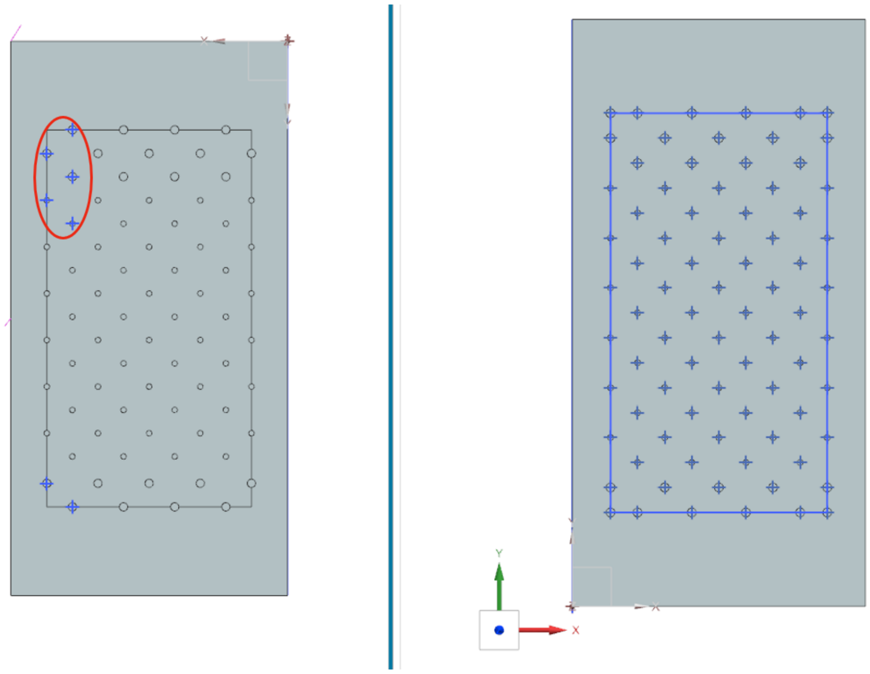


Figure 21. Two types of griding, the left one has a manual grid, right one has a diamond grid

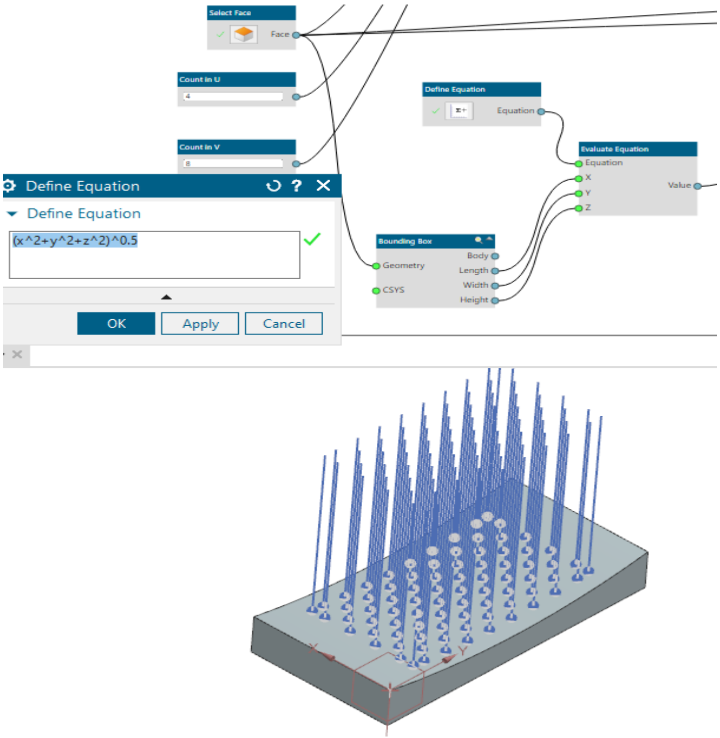
3.2 Creating the center line of the cylindrical shape for inserts

The location of the inserts is done by the holes feature and none as a selection for the Boolean for the points that are previously located by the way that is explained in section 3.1. The process continues by using pattern and mirror features. Some nodes might need to be changed for the length of the insert and diameter and the designer should check those points that if the diameter and length of holes need to be changed by putting the proper length according to the remaining back thickness. On the other hand, in the algorithmic feature, the only thing that is needed is to try to make points on the other surface of the 3D model. The equation for getting the point location for the other surface is done according to transferring the point from the insert surface to the other surface by the diameter of the face specified for the insert location. So, in this case below equation is used for transferring the points.

$$\sqrt{x^2+y^2+z^2}=r \qquad (3)$$

Equation (3) represents the radius of a sphere centered at the origin in three-dimensional space. This equation calculates the distance from the origin to any point (x, y, z) in space. So essentially, it gives the radius of a sphere that encompasses all points (x, y, z) such that their distance from the origin is the total distance. In this case, due to the flat surface as a face selection 680 mm×370 mm, the z value is 0 and the result would be around 774 mm.

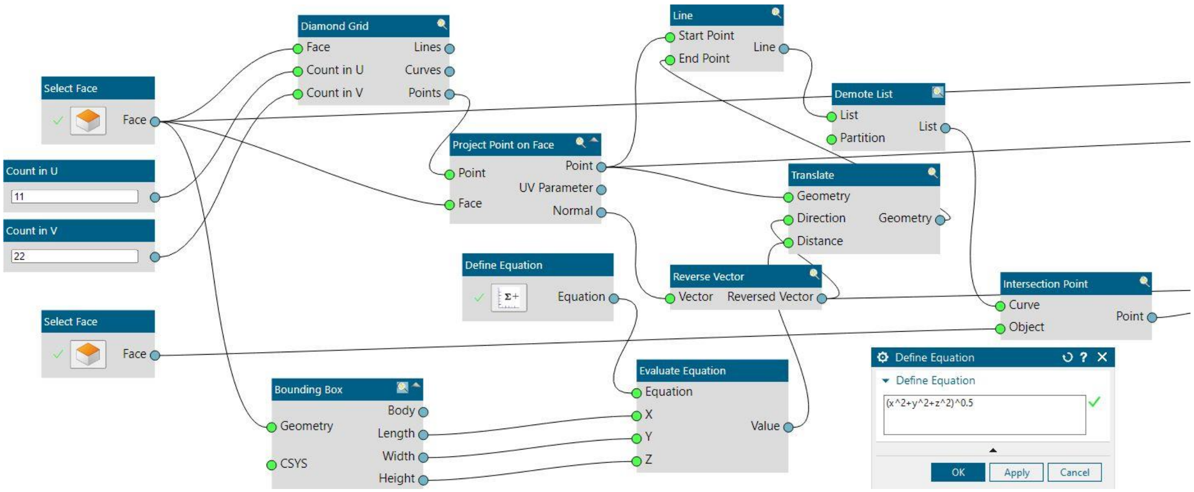
The intersection between those lines and the bottom surface creates the points on the back surface. In this way thickness for each point can be measured. Figure 22 shows how the points are translated to the other locations in space by using the diameter of the bounding box for the distance input in the Translate node and then creating a specific line for each point starting from the insert surface to the outer surface of the 3D model. Using the Demote list node to have the lines in different lists and using those lists for the input of the Intersection point node and the back surface of the 3D model as a face selection input for that node to get the points on the other surface of 3D model as shown in Figure 22. Figure 23 shows the algorithmic feature for getting those points on the other surface of the model.



(x^2+y^2+z^2)^0.5

Figure 22. Insert points transferring to the points in space

So according to Figure 23, the points on the back surface of the 3D model can be specified first using the face that inserts are located on as an input face for the Diamond grid. After which the project point on the face is used for creating a point on the face. Then those points should be replicated to the other surface using the translated node and Evaluate Equation as an input for that.



(x^2+y^2+z^2)^0.5

Figure 23. Nodes and algorithm for getting points on the other surface

3.3 Creating the line with insert surface points and the other surface points

After getting intersection points from touching the lines to the back surface, the point lists are created within the list, which means that the program analyzes each point intersection with the object face that was chosen as an input and the output would be the yellow line display data. Because of that, the list should be merged into one list again by using the Merge at the Level node. If that node is used, the list changes to the new list shown in Figure 24 with red line display data. In the next step, using the Flatten list can change the list in a way that all values do not have any lists anymore (blue line display data) and can be chosen for the input of the creating Line node which is the start point of that line is the output point from the project point on face.

Object	Index	Description
List	0	List with 1 Point objects.
Point	0	163.636363636, 320.000000000, 28.837272553
List	1	List with 1 Point objects.
Point	0	130.000000000, 350.909090909, 31.037225555
List	2	List with 1 Point objects.
Point	0	163.636363636, 381.818181818, 33.134372574
List	3	List with 1 Point objects.
Point	0	197.272727273, 350.909090909, 31.037225555
List	4	List with 1 Point objects.
Point	0	230.909090909, 320.000000000, 28.837272553
List	5	List with 1 Point objects.
Point	0	230.909090909, 381.818181818, 33.134372574

Object	Index	Description
List	0	List with 1042 lists and 521 Point objects.
List	0	List with 1 lists and 1 Point objects.
List	0	List with 1 Point objects.
Point	0	163.636363636, 320.000000000, 28.837272553
List	1	List with 1 lists and 1 Point objects.
List	0	List with 1 Point objects.
Point	0	130.000000000, 350.909090909, 31.037225555
List	2	List with 1 lists and 1 Point objects.
List	0	List with 1 Point objects.
Point	0	163.636363636, 381.818181818, 33.134372574
List	3	List with 1 lists and 1 Point objects.
List	0	List with 1 Point objects.
Point	0	197.272727273, 350.909090909, 31.037225555
List	4	List with 1 lists and 1 Point objects.
List	0	List with 1 Point objects.
Point	0	230.909090909, 320.000000000, 28.837272553
List	5	List with 1 lists and 1 Point objects.
List	0	List with 1 Point objects.
Point	0	230.909090909, 381.818181818, 33.134372574
List	6	List with 1 lists and 1 Point objects.

Object	Index	Description
Point	0	163.636363636, 320.000000000, 28.837272553
Point	1	130.000000000, 350.909090909, 31.037225555
Point	2	163.636363636, 381.818181818, 33.134372574
Point	3	197.272727273, 350.909090909, 31.037225555
Point	4	230.909090909, 320.000000000, 28.837272553
Point	5	230.909090909, 381.818181818, 33.134372574
Point	6	264.545454545, 350.909090909, 31.037225555

Figure 24. Creating a line from the insert surface points to the back surface points

3.4 Proper Insert Length

According to Table 1 and the feature of the insert's lengths, the nodes are used so that the back thickness meets the design requirements for two types of design. Figure 25 shows how back thickness is considered in algorithmic nodes for case 2 design according to Table 3. The first node that should be used for this purpose is the curve length which can measure the length of the line that has already been created in the previous section. Figure 25 shows one line that is signed by a rectangular cyan color. The cylinder has been already created with the proper length. The original line's length from insert surface to back thickness was around 107mm, signed by orange color. According to Table 2, the minimum back thickness is defined by the Minimum clearance with the inner node which is 20mm for the case 2 design.

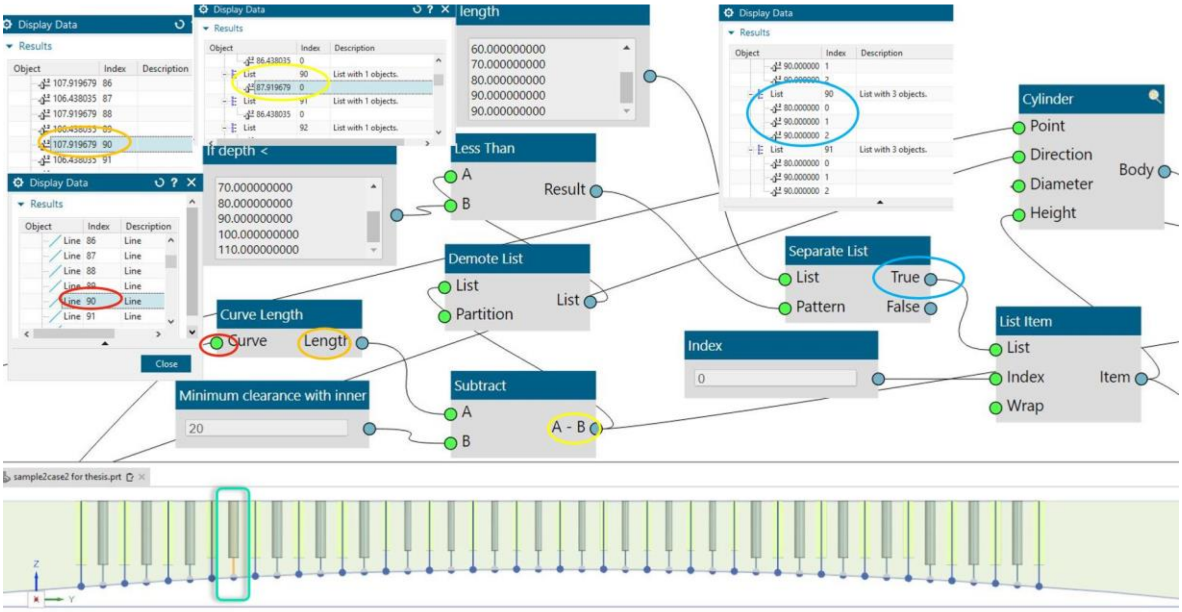


Figure 25. Algorithmic rules and model preview for the proper insert length

The subtract node is used to subtract the original length from 20 mm (A-B) which follows the Demote list node that can create the list for each point. Then the output goes through the Less Than node which compares the output value from the Demote list to the desired length that the designer specifies as the B input of the Less Than node. For instance, for the Line with index 90, the algorithmic nodes are working like the below flowchart, till three outputs are gotten from Less Than node which are 110 mm, 100 mm, and 90 mm value corresponds to 90 mm, 90 mm, and 80 mm value in length List which is the input for the Separate List node and after that getting those value as True output from Separate List node.

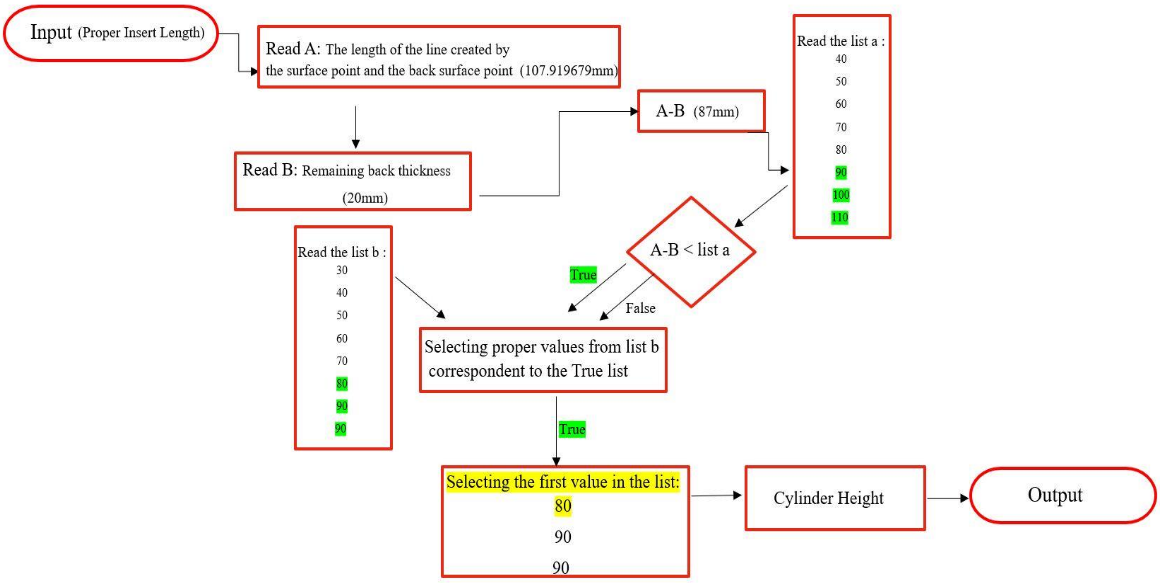


Figure 26. Flowchart for the proper insert length

Using a list item is crucial to select the proper insert length among those three True outputs separate list. Because the 80 mm length is required for the line 90, index 0 is chosen as an input for the List item node. If the 90 mm is chosen, the back thickness would not be the correct value (20 mm) and it will be around 18 mm, not according to Table 2 design requirements.

3.5 Proper inserts diameter

For selecting the proper diameter of the inserts, at some points looks like the previous method that is used for the length of the inserts but instead, the list of the diameters according to Table 1 is defined in the list of diameters (the list b in Figure 27). It follows the flowchart (Figure27) to get the two diameters value of 15mm and finally, one of them is selected.

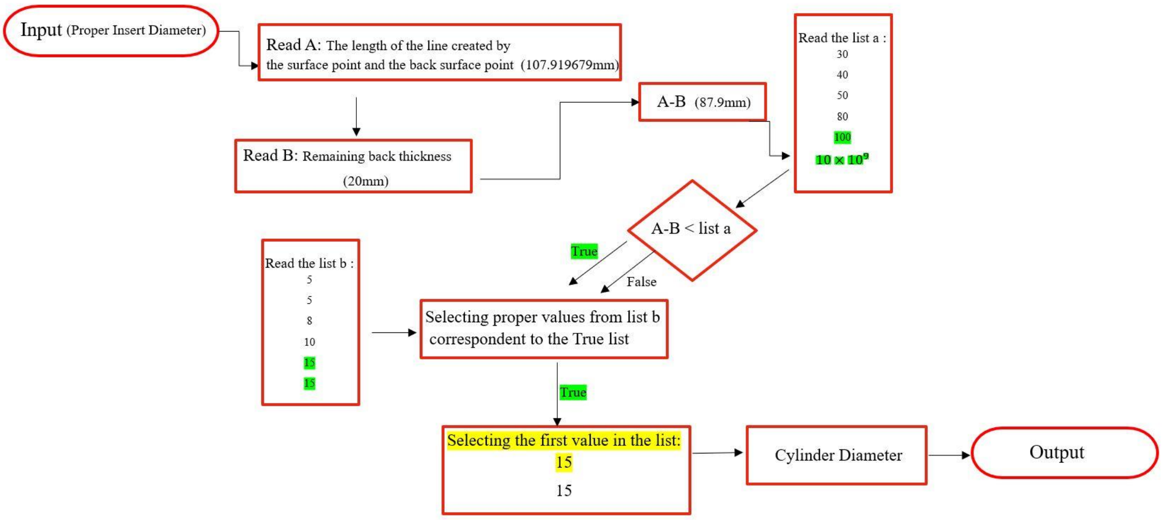


Figure 27. Flowchart for the proper insert diameter

The algorithmic nodes are going in a way that according to the proper length, the proper diameter will be selected. For instance, line 90 which is shown in rectangular cyan color in Figure 28 the length of that line is 107.919679 mm (orange color), subtracting by 20 mm it is changed to 87.919679 mm after which it goes through the demote list to put that point in specific list number to compare with the other values that are in the list If depth <.

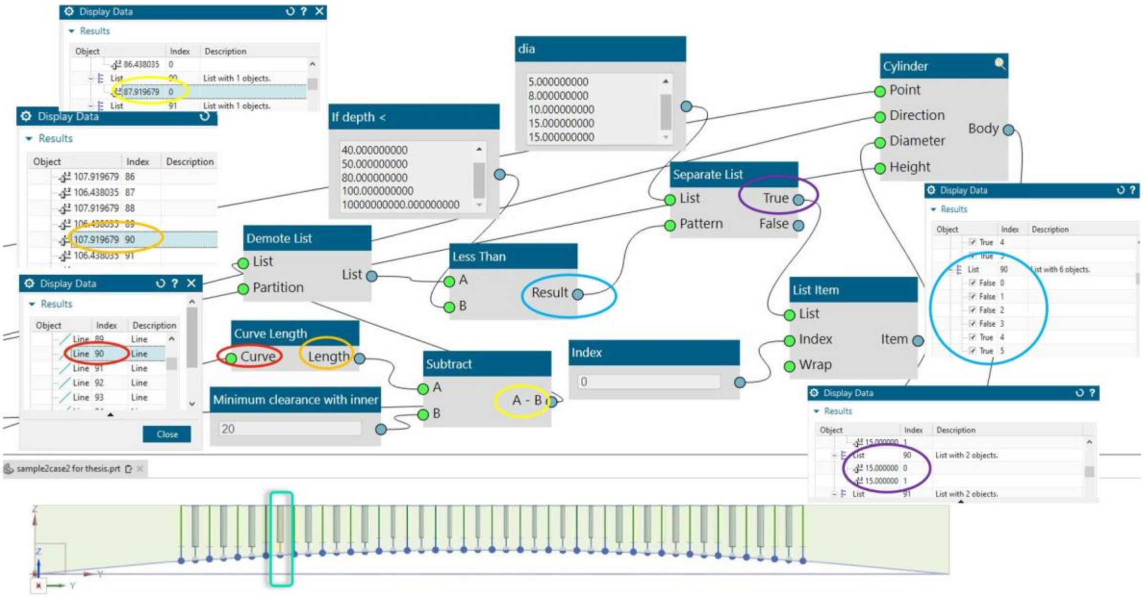


Figure 28. Algorithmic rules and model preview for the proper insert diameter

after comparison with the list (If depth<) values, among 6 items, two of them are true which shows that those are greater than 87.9 mm (blue circle). to get the proper insert length corresponding to the If depth< values, the separate list node is utilized for selecting the 15mm diameter. To have the one insert diameter according to the correct diameter size, the List item node is used, and the input index is 0 for that. So, line 90 has the cylinder feature which is inserted on that location that has 80 mm length and 15 mm diameter.

3.6 Inserts distribution

One model design element that is connected to insert density on the face is insert distribution. The insert distribution is about the spacing between inserts and the layout of the inserts. A surface with a high insert density has little spacing between inserts, and vice versa. Insert destruction may also be impacted by the patterning technique and form.

3.6.1 Adjusting the density of inserts on the surface by algorithmic feature

Inserts' density can be changed in algorithmic feature by changing the U and V parameters from the input of the diamond grid available in the dial box for algorithmic rules. Designers can make decisions to use the best grid method from different types of grids in algorithmic feature to get the proper distance. In this design case, this change can be done in less than a second.

3.6.2 User expressions for adjusting the density of the inserts in manual design

Insert density refers to the distribution and spacing of inserts on a surface body, which affects features of the product such as lifetime, weight, and overall performance. This modification is normally performed by changing the positions of components on the surface. Designers used to manually size the distances between such factors, which often took a long time and involved a lot of trial and error. However, designers can use user expressions to speed up this process.

3.7 Checking process in manual design

After creating 5 points on the surface designer should use the hole feature to get the insert shape and according to the diameter and length which are available in Table 1, the model can be modified to reach the minimum back thickness of 30mm in case 1 design and 20mm for the case 2 design. After that, the designer should control the spacing between the inserts that are created. If any changes need to be made, it should be done from the first step by controlling the distance by modifying the user expression for the location of the points, which is somehow time-consuming.

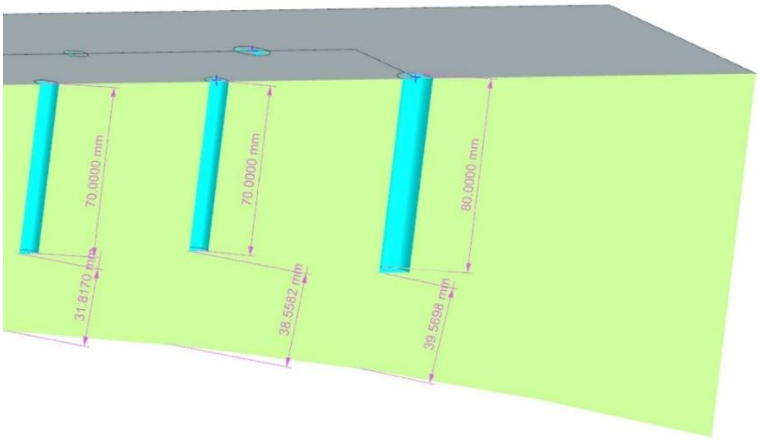


Figure 29. Minimum back thickness checking in manual design case 1

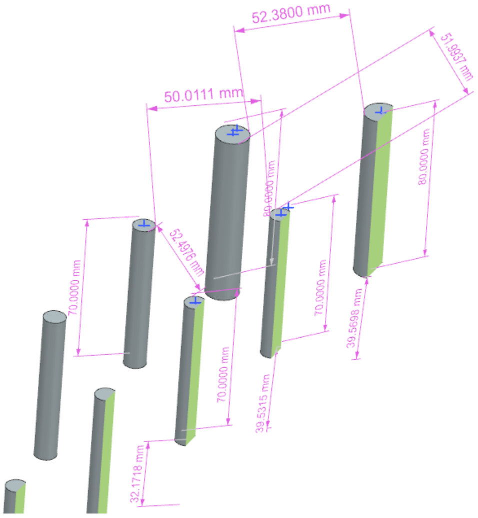


Figure 30. Checking the design requirements according to Table 1

In Figure 31, the insert location for the manual model is illustrated. All the criteria for design are according to Table 1 and the total time for putting inserts for the design is roughly around 30 minutes.

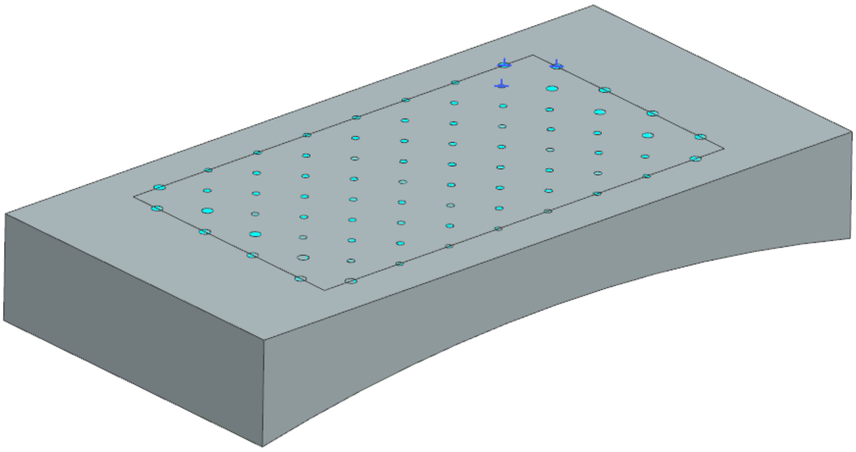


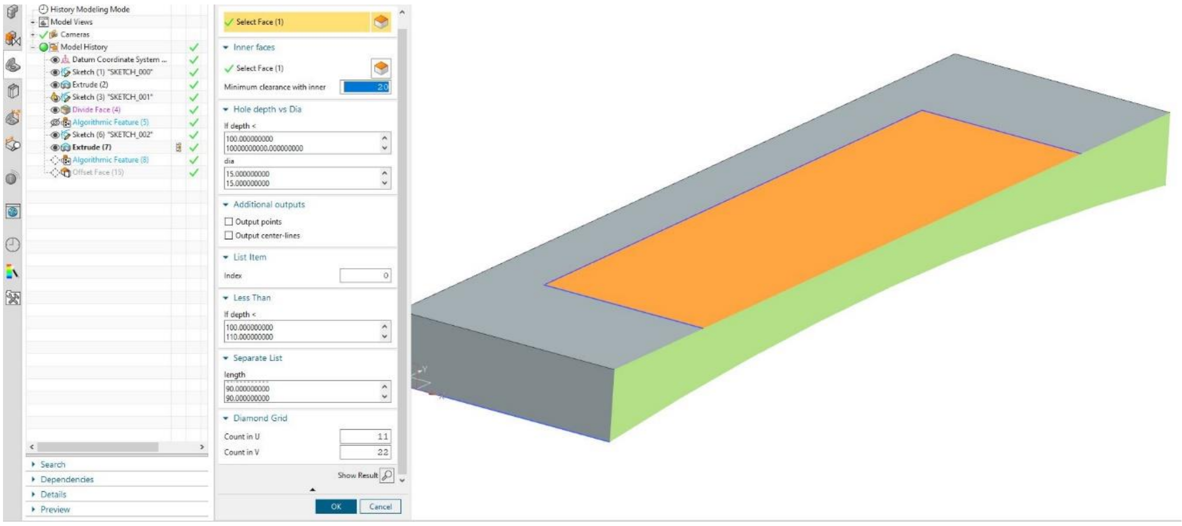
Figure 31. Case 1 design model by using manual and user expression for insert points

3.8 Checking process in algorithmic design

The user can specify and change parameters that are essential for design purposes with the help of an algorithmic feature. The designer should review the design requirements throughout the process. Initially, the criteria can be defined as an input in the dialog box for the algorithmic feature.

3.8.1 Algorithmic feature for design purpose

The initial step in algorithmic design can be time-consuming since it needs careful thought of how to develop rules and connect nodes to obtain the desired design. During the first few attempts, this stage could last hours, weeks, or even months. Once the algorithm is completed, it can be used for different models with various sizes. Designing the complex patterning feature for case 1 design is conducted according to the design details and Table 1 for insert length and diameters. The algorithmic is created within one week for that design purpose and the designer can make the changes in the NX user interface without even going and editing the basic rules and nodes in the logic editor. Designers can make the auto-generated dialog box easier and friendlier by editing it from the Rule Inputs of Logic Editor. In Figure 30 dialog box is shown in which designer can edit and change the design criteria in the modeling environment and avoid going through all nodes and connectors in the algorithmic feature.



Select Face (1)

Inner faces
Select Face (1)
Minimum clearance with inner: 20

Hole depth vs Dia
If depth <
100.000000000
10000000000.000000000
dia
15.000000000
15.000000000

Additional outputs
☐ Output points
☐ Output center-lines

List Item
Index: 0

Less Than
If depth <
100.000000000
110.000000000

Separate List
length
90.000000000
90.000000000

Diamond Grid
Count in U: 11
Count in V: 22

Figure 32. Dial box for editing design criteria in algorithmic design

3.8.2 Effective parameter in algorithmic design

In Figure 30, the dial box shows the effective parameters and by changing those parameters, the design will be changed according to product requirements. Minimum back thickness can be defined, length of inserts and diameter also can be modified by the designer in the dial box. U and V parameters also have an impact on griding and insert distance (minimum thickness between inserts).

3.9 Design time analysis for Case 1

To determine how much time is required for model one with the design requirement according to Table 2, the design time for model one using both algorithmic and manual approaches is analyzed.

3.9.1 Design Case 1 with Algorithmic

Total designing time using the algorithmic feature at first needs a quite bit of effort to get to know the best rules and the way to use the proper nodes and connectors. For the case 1 design, it takes 1 week to create the algorithmic feature for that shape of the model, and it can be expanded to any other size. After that, the total time for modifying the design is less than 10 minutes to complete and check that all the values in the dial box are all right for the design requirements like the list of inserts and the list of diameters that are included in Table 1.

3.9.2 Design Case 1 with manual

The total designing time for this method specifically for case 1 needs more time and takes 30 minutes at least to get the design ready although, for some points, some design parameters should be changed according to the back thickness, and the pattern feature cannot be useful. For specific points, designer might use a hole feature with a specific length and diameter.

3.10 Design time analysis for Case 2

3D models are analyzed in two ways. One with the manual design which needs quite a bit of time to get the design ready according to the design detail in Table 3. Figure 31 is both design illustration which shows the faster way of using the algorithmic feature in design compared to the manual and using the user expression. It is shown the short design tree in parametric design which affects the time of processing during finalizing the model.

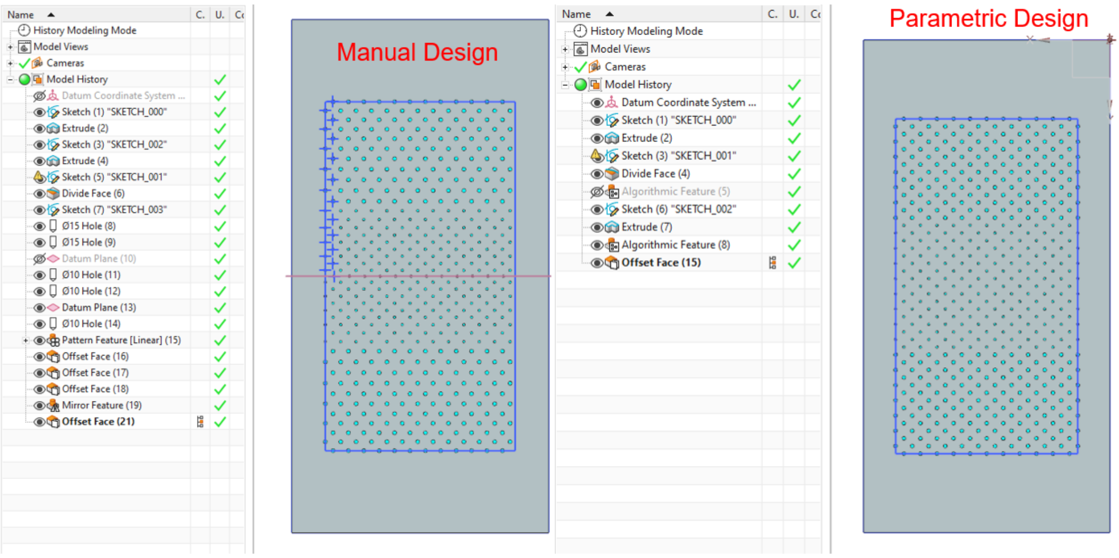


Figure 33. Complex patterning on the surface according to Table 3 requirements

3.10.1 Design Case 2 with Algorithmic Method

Design case 2 is analyzed in this section when the algorithmic feature is used and controls the design requirements and the modeling sizes in terms of length, width, thickness, and remaining back thickness, the model in design case 2 is bigger than case 1. However, using algorithmic features reduces the time of design, and getting done in 15 minutes just for checking the proper distance is implemented to get the insert spacing to be 30-35mm and the remaining back thickness to be 20mm. It is done without any error and smoothly is applied for the model although, in the dial box, the designer has changed the remaining back thickness requirement and modified the length and proper length of inserts in the dial box.

3.10.2 Design Case 2 with manual method

Comparing the time of editing in manual design with algorithmic one in design case 2 model, which has a larger model compared to case 1 and different design features, it is so time-consuming to editing that and check the distance and put proper inserts on the surface. Figure 32 shows how design case 2 is done by using user expression for two rows in the left corner and then using the pattern and mirror feature to get the result and suitable design. It takes 60 minutes to do that complicated patterning insert on the 3D model surface.

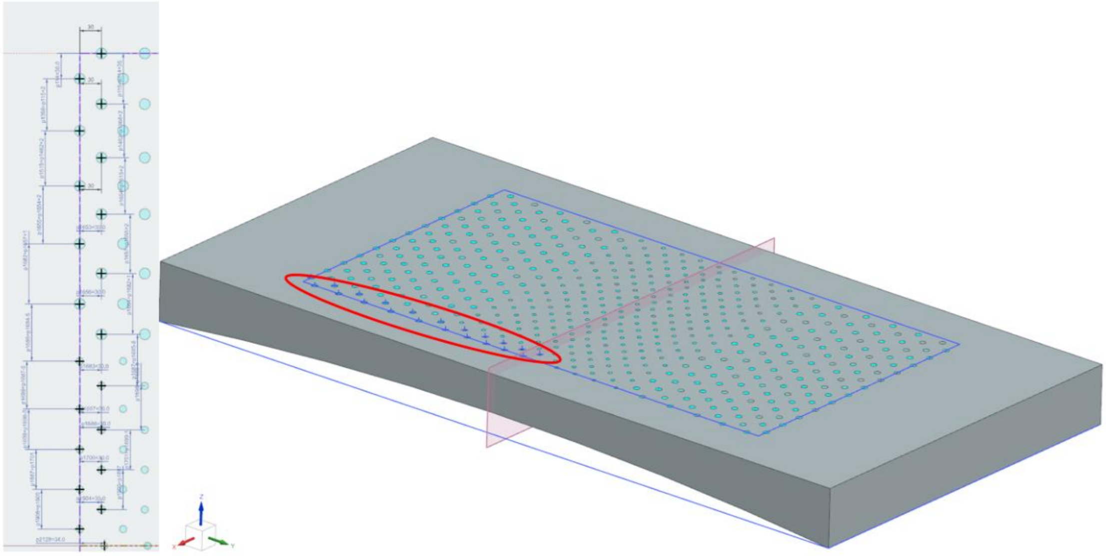


Figure 34. User expression usage for insert location in manual design case 2

3.11 Design time analysis in 6 cases

Two cases were conducted using two different methods, manually and parametric, which in total contain 4 designs, two of them are manually and the other one is parametric. In manual design, the first part which has a small geometry is done in 30 minutes shown in Figure 33. However, the other second design, which is done manually, takes 60 minutes roughly due to the increasing geometry of the 3D model in length and width. The time it takes to develop complex patterns increases as the complexity of the model grows for example, from Case 3 (4000mm length and 2000mm width) to Case 4 (8000mm length and 4000mm width). This process continues until we reach Case 6. Designing those intricate designs takes 960 minutes by that point. Overall, it takes 1890 minutes for all those 6 cases if they are done manually. On the other hand, although the baseline for the algorithmic feature needs time and roughly it is like 1500 minutes (5 working days), it is so promising cause after taking much time at the beginning of the project it can reduce the time of each design case even if the shape of the geometry is changed and increasing double time. Showing 1575 minutes for 6 different sizes of the model with one baseline for creating algorithmic features still can have a good impact of reducing the total time of design by using the parametric method.

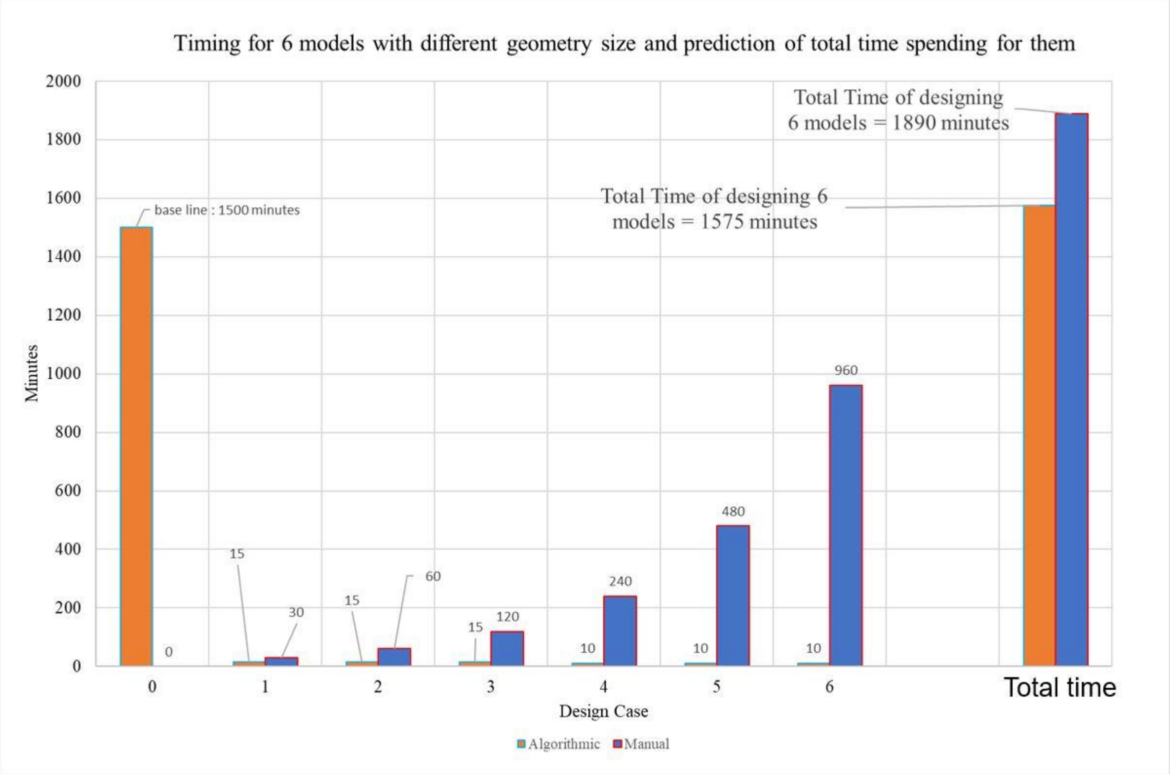


Figure 35. Design time analysis

3.12 Design cost for six case studies

According to the reduction of the design time, the cost of that is affected by using faster methods in the project which has different types of design. If the designer can use the algorithmic feature for different models, then the total amount of money that needs to be spent for that model is going to be reduced. For the previous analysis of 6 designs, Table 4 shows the benefit of using algorithmic features instead of using conventional methods considering 40 euro per hour for the design cost.

Table 4. Design cost analysis

Design Approach	Total Minutes	Total Hours	Total Cost (€)	Cost Reduction (%)
Algorithmic	1575	26.25	1050	-16.67
Manual	1890	31.50	1260	

3.13 Advantages of Algorithmic Design

In the first design step, using algorithmic features sounds a little bit complicated and needs a lot of time to figure out the best options and methods to use the nodes and connectors and apply rules in a way that meets the design requirement. However, after that, the algorithm can be expanded to all other geometry shapes that have a specific design feature. Like in complicated patterning items, the algorithmic rules can apply to the different shapes even if the shape is conical or flat with curved edges and faces at some point.

3.13.1 Simulation Time and Flexibility

Designers benefit from the flexibility inherent in algorithmic design, which saves significant time as compared to manual approaches. Instead of starting with hand sketches to outline patterning characteristics, designers can use parametric features, which not only speed up the design process but also allow them to explore numerous mathematical solutions through coding, such as using NX Open. This capability enables designers to create rules that improve operational efficiency, making the design process faster and more dynamic.

The time of simulations is directly proportional to the complexity of the rules and the number of components, such as boxes and connectors, included in each feature. When used properly, simulation times remain reasonable for design purposes. Although constructing the basic reasoning for a new design may take some time, once established, this logic can be applied across many shapes and geometries, resulting in significant time savings when developing models of varying sizes and configurations. Finally, this approach allows designers to produce consistent results across varied shapes while adhering to the same design standards.

3.13.2 Design Improvement and Skill Ability

Skill ability and improvement are other advantages of this feature. Designers can improve their ability in this feature expand their knowledge in this field and use it for the 3D model with different geometry which has the same design criteria. With algorithmic design, designers can easily handle complex geometries. Algorithms provide the tools necessary to construct detailed designs precisely and effectively, whether they are complicated patterns on curved surfaces or flat surfaces. Algorithmic design gives a stable basis that may be applied to a variety of shapes. This means that designers can apply the same ideas to different geometries while remaining consistent and coherent in their creations. The designer’s skill can be improved in this way and can have a good impact on the complex design that the designer might face challenges design in the future.

4 Conclusion

The revolutionary potential of algorithmic techniques in modern design practices is examined in this thesis, with a focus on Metso's 3D development of crusher wear products. First, parametric modeling is discussed in the framework of a survey of the literature. Metso's crusher wear designs require complex patterning, which makes it necessary to use generative design, a subset of parametric design made possible by visual programming. With the inherent algorithmic capabilities of 3D modeling tools, generative design provides a way to deal with this kind of complexity. The process entails breaking down algorithms to find important nodes and characteristics that are necessary to fulfill certain design specifications. This gives a general idea of how generative design might be applied to different Metso products.

The thesis's basic design was based on a particular 3D model that was chosen to be similar to Metso's crusher wear product. A griding node within the algorithmic feature was utilized to indicate the center locations of inserts to generate the complex patterning on its surface. Each insert's length and diameter followed algorithmic rules, making it simple for designers to modify to suit their requirements. This technology is transferable to other geometries with comparable patterning properties. Through the advent of reusable logic and parameters, designers can explore a huge variety of features within shorter timeframes, in the long run reducing design times and increasing productivity. Moreover, algorithmic features foster flexibility and customization by permitting dynamic changes to design parameters. Designers can effortlessly tailor designs to satisfy unique requirements by clearly adjusting input parameters, resulting in surprisingly customizable and adaptive designs.

The algorithmic functionality was used to make it easier to compare the design timing of two models, demonstrating its effectiveness in comparison to manual methods. Advantageous design timings were noted with the algorithmic feature, which is especially helpful for designers who want to maintain design consistency amongst geometries that have the same design requirements. The algorithmic method showed a total design time of 1575 minutes over six design scenarios with different insert combinations and dimensions, while the manual method took 1890 minutes. In addition to saving money on design, this time-saving feature can be applied more broadly, improving the skill of designers and allowing algorithmic features to be reused in other organizational projects.

In the future, research on algorithmic design and visual programming in the field of design may be further improved. It is suggested that investigating the theoretical coding and programming of individual nodes inside the algorithmic feature should broaden knowledge and skills, considering the paucity of study in this field. Combining this knowledge with the visual basic code in NX Open offers a fruitful study direction. Further research could focus on improving algorithmic feature design using NX Open, customized griding in algorithmic features using coding and adding to algorithmic rules, and theoretical analysis of the algorithmic feature in the design of complex patterning features in 3D models. Through these initiatives, the area of design is encouraged to be more innovative as algorithmic design knowledge and techniques are advanced.

References

Abdullah, H.K. and Kamara, J.M., 2013. Parametric design procedures: a new approach to generative-form in the conceptual design phase. In AEI 2013: Building Solutions for Architectural Engineering (pp. 334-343).

Ajouz, R., 2021. Parametric design of steel structures: Fundamentals of parametric design using Grasshopper. Steel Construction, 14(3), pp.185-195.

Aranburu, A., Camba, J.D., Justel, D. and Contero, M., 2023. An Improved Explicit Reference Modeling Methodology for Parametric Design. Computer-Aided Design, 161, p.103541.

Castelo-Branco, R., Caetano, I. and Leitao, A., 2022. Digital representation methods: The case of algorithmic design. Frontiers of Architectural Research, 11(3), pp.527-541.

Cucos, M.M., Pista, I.M. and Ripanu, M.I., 2018. Product engineering design enhancing by parameterizing the 3D solid model. In MATEC Web of Conferences (Vol. 178, p. 05011). EDP Sciences.

Jiao, L.L., 2015. Design of valve parametric variant template based on NX. Applied Mechanics and Materials, 697, pp.289-292.

Jaisawal, R. and Agrawal, V., 2021, March. Generative Design Method (GDM)–a state of art. In IOP Conference Series: Materials Science and Engineering (Vol. 1104, No. 1, p. 012036). IOP Publishing.

Jang, S., Yoo, S. and Kang, N., 2022. Generative design by reinforcement learning: enhancing the diversity of topology optimization designs. Computer-Aided Design, 146, p.103225.

Kravchik, T., 2023, April. CAM model parameterisation methodology and its further unification in Siemens NX environment. In E3S Web of Conferences (Vol. 389, p. 01009).

Kyratsis, P., Manavis, A. and Davim, J.P. eds., 2023. Computational design and digital manufacturing. Springer Nature, p. 205–216.

Li, H. and Lachmayer, R., 2018, August. Generative design approach for modeling creative designs. In IOP Conference Series: Materials Science and Engineering (Vol. 408, No. 1, p. 012035). IOP Publishing

McKnight, M., 2017. Generative Design: What it is? How is it being used? Why it'sa game changer. KnE Engineering, pp.176-181.

Novak, J. and Loy, J. (2017) 'Recoding Product Design Education: Visual Coding for Human Machine Interfaces', KnE Engineering, 2(2), p. 227.

Rawat, A.S. and Tiwari, G., 2022, August. Modern Generative Design Tools: Siemens NX's Algorithmic Feature and Rhinoceros 3D's Grasshopper. In Biennial International Conference on Future Learning Aspects of Mechanical Engineering (pp. 275-284). Singapore: Springer Nature Singapore.

Stroud, I. and Nagy, H., 2011. Solid modelling and CAD systems: how to survive a CAD system. Springer Science & Business Media. Pp. 53-63.

Sun, B. and Huang, S., 2019, July. Realizing product serialization by Grasshopper parametric design. In IOP Conference Series: Materials Science and Engineering (Vol. 573, No. 1, p. 012078). IOP Publishing.