POLITECNICO DI MILANO

Master in

Building Information Modelling

BIM A+ European Master in Building Information Modelling

(Optimizing bridges modelling through visual programming)

(Ottimizzare la modellazione di ponti tramite la programmazione visuale)

Supervisor:

Professor Claudio Mirarchi

Author:

Sayed Mahdi Mousavi

Co-funded by the Erasmus+ Programme of the European Union

2022/2023

AUTHORSHIP RIGHTS AND CONDITIONS OF USE OF THE WORK BY THIRD PARTIES

ACKNOWLEDGEMENTS

I would like to express my gratitude to all who have helped me in this way. I would like to thank Professor Claudio Mirarchi, my supervisor, and Paola Erba, the chairman of Alpina, for their great guidance and priceless knowledge. For their advice and support, Marco Bonfanti and Marco Ghidoli at Alpina have my sincere gratitude. I would like to express my sincere gratitude to all the professors who have guided me during my academic career, the BIM A+ coordination, my family, my beloved academic peers, and my friends. Thank you for the financial assistance from the consortium scholarship.

I would like to present this dissertation to the girls of my country who are prohibited from studying and working.

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of ethics and conduct of Politecnico di Milano.

SOMMARIO

Vari ambiti dell’ingegneria hanno fatto progressi significativi grazie all’uso della programmazione visuale e del Building Information Modeling (BIM). In questo studio, ci concentreremo sull'utilizzo di strumenti di programmazione visuale per migliorare le procedure di modellazione di ponti e affrontare i problemi legati all’uso di approcci tradizionali.

La questione principale sollevata in questa tesi riguarda i limiti dei metodi convenzionali di modellazione dei ponti che limitano l’efficacia dell’analisi e della progettazione strutturale. L'obiettivo è sfruttare appieno le capacità della programmazione visuale per trasformare le procedure di modellazione di ponti e accelerarla.

Gli obiettivi di questo studio sono affrontati combinando tecniche di ricerca qualitativa e quantitativa. Sono state effettuate valutazioni per comprendere l'attuale stato dell'arte nella modellazione di ponti e i possibili vantaggi della programmazione visuale. Per acquisire informazioni approfondite da progetti reali di ponti e consentire una valutazione approfondita delle tecniche proposte, vengono utilizzati anche casi studi e interviste agli esperti.

I risultati di questo studio indicano come la programmazione visuale abbia un enorme potenziale per migliorare i modelli di ponti. È possibile utilizzare uno script Dynamo per generare il ponte completo in una frazione del tempo necessario alle tecniche umane, riducendo i tempi del processo di modellazione da settimane a pochi secondi. Questa generazione rapida consente di apportare modifiche rapide alla progettazione del ponte, inclusa la regolazione della campata e dell'altezza della pila. Inoltre, l'uso della programmazione visuale accelera la procedura di analisi strutturale, consentendo ai progettisti di scegliere con sicurezza, per esempio, la migliore trave in calcestruzzo per un ponte in calcestruzzo.

Parole chiave: (BIM, Ponti, Dynamo, Ottimizzazione, Revit)

ABSTRACT

Various engineering fields have significantly advanced because of the use of visual programming and Building Information Modeling (BIM). In this study, we concentrate on using visual programming tools to improve the bridge modeling procedure and deal with problems caused by traditional approaches.

The main issue raised in this thesis is the limits of conventional bridge modeling methods that limit effective structural analysis and design. The objective is to fully utilize visual programming's ability to transform the bridge modeling procedure and speed up.

The goals of this study are addressed by combining qualitative and quantitative research techniques. We undertake evaluations to comprehend the current state-of-the-art in bridge modeling and the possible advantages of visual programming. To acquire insightful information from actual bridge projects and enable a thorough assessment of the suggested technique, case studies, and expert interviews are also used.

The findings of this study indicate how visual programming has enormous potential for improving bridge models. A Dynamo script may be used to generate the complete bridge in a fraction of the time needed by human techniques, cutting the modeling process from weeks to just a few seconds. This quick generation makes it possible to make quick changes to the bridge design, including adjusting the span and pier height. Additionally, the use of visual programming speeds up the structural analysis procedure, enabling designers to choose e.g., the best concrete girder beam for a concrete bridge with confidence.

Keywords: (BIM, Bridges, Dynamo, Optimization, Revit)

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

1.1. Background

Modern infrastructure is not complete without bridges, which link towns and facilitate the effective movement of people and products. Complex engineering factors, such as structural integrity, load analysis, material selection, and aesthetically pleasing design, are taken into account while designing bridges. Traditional bridge modeling techniques frequently rely on specialist software with a more difficult learning curve and a linear approach to design changes. The design of building facilities and transportation infrastructure is increasingly utilizing BIM (Building Information Modeling) techniques, and in certain situations, it is even required (Del et al, 2017).

The development of visual programming has altered the way that engineers and designers approach difficult projects like bridge modeling. Using graphical interfaces and blocks or nodes to express design and logic, visual programming enables users to build, manipulate, and analyse complicated models. This paradigm change in modeling creates chances to expedite the design process, boost creative thinking, and make multidisciplinary teams more collaborative.

Instead of depending entirely on conventional text-based coding, visual programming entails designing software programs employing graphical components and visual representations of code logic. By enabling developers to construct and edit code through visual components like blocks, nodes, and diagrams, visual programming languages (VPLs) promise to streamline the software development process. These programming languages frequently include drag-and-drop user interfaces, allowing users to choose and link visual components to specify the logic of their applications. This method can be very helpful for individuals who are new to programming because it lowers the entry barrier and makes it easier for them to understand programming ideas.

1.2. Scope of study

A full 3D model of a building or infrastructure project is created and managed digitally as part of the Building Information Modeling (BIM) process. It covers a structure's structural and functional attributes and acts as a shared knowledge resource for data regarding the project's lifecycle, from conception and design to building, use, and maintenance. However, the majority of building operation and maintenance procedures still use outdated methods and seldom rely on digital information.

BIM makes it easier for the different parties involved in a construction project, including architects, engineers, contractors, and owners, to collaborate. They can communicate, collaborate, and better coordinate their efforts by using the same 3D model. The core component of BIM is the creation of a 3D digital model of the infrastructure or building. This model's inclusion of both geometric and non-geometric data enables a more precise and thorough depiction.

Bridge design and construction reflect complex engineering problems that require careful planning. Technology development over years has had a considerable impact on the modeling and design of bridges, enhancing their sustainability, accuracy, and safety. In this thesis, we explore new techniques to improve the bridge design process while optimizing bridge models through visual programming.

The main issue in this study addressed with the limitations of bridge modeling methods that are typically hard-working and time-consuming manual processes. We postulate that Dynamo will change the design, and optimization of bridge models. The use of visual programming, which enables engineers to build complex algorithms and models using a graphical interface, has the potential to speed up the process, facilitate decision-making, and eventually result in the creation of cost-effective.

The purpose of this thesis is the use of visual programming tools in the context of bridge modelling. The following are the study's main goals:

a. To find out the current difficulties and constraints related to using traditional bridge modeling techniques.

b. to evaluate the potential and capabilities of visual programming tools in the context of bridge design and optimization.

c. To create best practices and recommendations for applying Dynamo in engineering projects.

Earlier studies in the field of bridge engineering and modeling have mostly concentrated on improving analytical and numerical techniques. Studies that investigated how Building Information Modeling (BIM) was included into bridge design showed advantages in communication and data sharing. In the academic literature, however, visual programming's application to bridge modeling is still largely unexplored. By highlighting the opportunities and difficulties of this novel method, this thesis seeks to add to the body of existing knowledge.

This study's research technique is based on a combination of qualitative and quantitative methods. To learn more about state-of-the-art bridge modeling and the possible advantages of visual programming, literature reviews, case studies, and expert interviews will be undertaken. In addition, actual bridge projects will be chosen as case studies to compare conventional modeling approaches with visual programming paradigms, resulting in a thorough assessment of their performance. Building information modeling offers advantages such as enhanced visualization, streamlined construction schedules, cost estimation, and better facility management. It also reduces errors and rework. Because it has the ability to boost collaboration, cut costs, and promote efficiency, BIM usage has been gradually increasing in the construction sector. Although there may be a learning curve for individuals unfamiliar with the technology, its successful deployment necessitates strong communication and cooperation among all project stakeholders.

1.3. Dissertation outline

Bridges play a crucial role in the infrastructure, ensuring connectivity and accessibility in many areas. However, the traditional approaches to modeling and optimization of bridges are sometimes burdened with complications that obstruct effective design and resource allocation. In order to streamline the modeling process, this dissertation uses visual programming to revolutionize bridge engineering. In order to develop a streamlined process for bridge modeling and optimization, the project intends to take advantage of the adaptability and user-friendliness of visual programming platforms, such as Grasshopper for Rhino or Dynamo for Revit. The project aims to show how visual programming may surpass the constraints of conventional approaches, encouraging more precise and resource-efficient bridge designs. It does this by merging these platforms with cutting-edge optimization algorithms.

It will clarify the difficulties that come with traditional methods, such as computer-aided design and finite element analysis, and point out their drawbacks in terms of complexity, duration, and accuracy. Additionally, the review will delve into the field of optimization in bridge engineering, looking at various methods including genetic algorithms and particle swarm optimization. The research will also explore the expanding uses of visual programming in civil engineering and related fields. The dissertation's goal is to provide the groundwork for a novel strategy that optimizes bridge modeling operations through visual programming, ultimately resulting in improved bridge design effectiveness and structural performance.

This page is intentionally left blank

2. METHODOLOGY

2.1. Research design and methodology

It is theoretical research. By graphically showing impossible coding procedures, visual programming frequently makes them simpler. With less need for in-depth coding knowledge, engineers, architects, and other experts may find it simpler to develop, change, and improve bridge models. Engineers can easily prototype and see various bridge design iterations thanks to visual programming tools. The optimization of structural components and configurations may be more effective because of this rapid iteration.

Engineers can establish relationships between variables and parameters using parametric design, which is supported by many visual programming systems. Changing parameters and monitoring changes to the model's behavior and performance in real time can assist in automating the optimization process. Various data sources, including environmental information, material characteristics, and load data, can be coupled to visual programming interfaces.

Engineers can make better choices about tailoring the bridge for particular situations by incorporating data into the design process. The creation of accurate and efficient models requires the use of a variety of data types, which are crucial for optimizing bridge modeling using visual programming. This information covers a broad range of topics, such as bridge specifications, structural characteristics, material characteristics, environmental factors, and more.

2.2. Optimization algorithms

Define the optimization issue you wish to fix first. This could entail choosing design parameters that will be susceptible to optimization, such as material qualities, dimensions, and structural components. Indicate the goals you want to achieve, such as reducing the amount of materials used, the cost of building, or increasing structural performance indicators like load-bearing capability. Transform the necessary design variables into a form that the optimization method can work with. These criteria could relate to the bridge's geometry, material characteristics, and other design considerations.

The choice of algorithm is influenced by elements including problem complexity, convergence rate, and solution quality. Each algorithm has strengths and limitations. Grasshopper for Rhino or Dynamo for Revit are two examples of visual programming environments in which you may embed the optimization method. In order to integrate the optimization method and the bridge modelling process, a workflow must be created. The optimization algorithm's inputs and outputs should be connected to the appropriate visual programming canvas elements.

The design variables will be iteratively adjusted by the optimization method to converge on an ideal or nearly ideal solution. In each iteration, a fresh bridge model is created, its fitness is assessed, and the design variables are revised in accordance with the algorithm's optimization strategy. Record every step of the optimization process, such as the algorithm selection, parameter settings, convergence findings, and the final optimized design. Within the context of bridge modeling and your research goals, discuss the consequences of the optimization results.

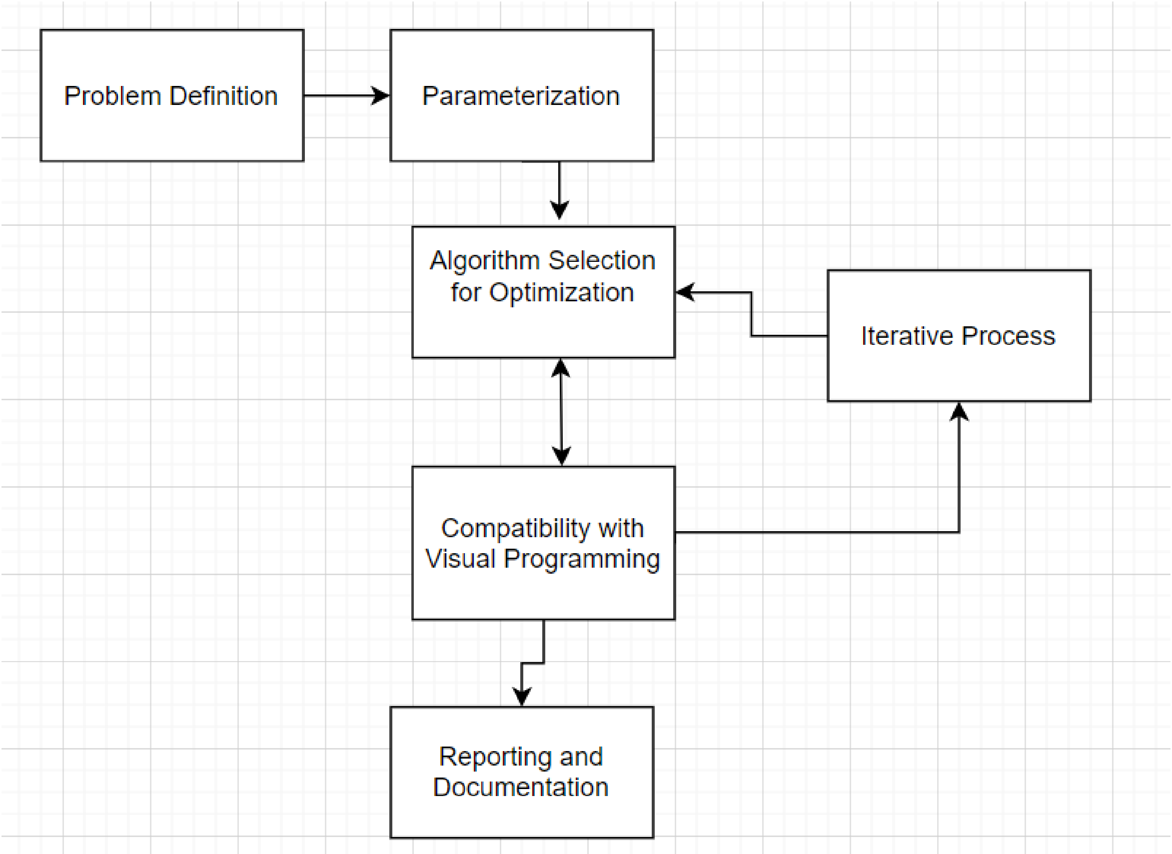


Figure 2.1 – (Implementation procedure for optimization algorithms)

2.3. Data gathering and the implementation of optimization techniques

It is possible to get this data using a variety of techniques, including simulations, field observations, and the use of existing datasets. In the following there are many methods to collect information. Dimensions, span lengths, clearances, piers, abutments, and other geometrical details of the bridge, parameters pertaining to the bridge's structural elements, such as the size of the beams, the height of the columns, the specifics of the reinforcing, etc. taken from the old drawing sheets. In this research, Revit and Dynamo are used to model the bridge.

The output of this research is to model the entire bridge in a short time by the help of Dynamo and Revit. The same idea, but not the same script, it could be possible to model other elements like pipeline network in a short time.

The specified visual programming platform must be integrated with the desired optimization techniques. This could entail writing unique scripts or components to reflect the optimization procedures. The seamless execution of optimization activities within the bridge modeling environment is made possible by this integration.

Gather the data that will be needed as input throughout the optimization procedure. The geometry of the bridge, the materials, the loads, and any design factors might all be included in this data. In order to enter the data into the visual programming environment, be sure it is correctly formatted. Create a visual programming workflow that incorporates the bridge modeling method and the optimization algorithm.

The method iteratively investigates various design options, modifying design variables to fulfill limitations and meet the given objectives. The objective function is assessed on each iteration, changes are made, and the cycle is repeated until convergence is reached.

This page is intentionally left blank

3. BRIDGE MODELING TECHNIQUES AND VISUAL PROGRAMMING

3.1. Bridge modelling techniques

A Building Information Modeling (BIM) environment can be efficiently used with bridge modeling techniques in Revit to produce accurate and detailed representations of bridge structures. By allowing engineers and designers to seamlessly combine bridge design and analysis, they can facilitate a streamlined workflow and improve teamwork across disciplines. The parametric modeling strategy is one of the core components of bridge modeling in Revit. For numerous bridge elements, including piers, abutments, girders, and decks, engineers can develop flexible parametric families. Users can quickly change dimensions and characteristics to investigate various design possibilities while keeping consistent updates throughout the entire model by defining linkages between parameters.

The application of adaptable components is a crucial technique. These elements make it possible to create complicated and unusual bridge geometries, like curved alignments and asymmetrical shapes. The bridge model can more correctly adapt to site conditions in the real world thanks to adaptive components, resulting in a better level of representational precision.

Revit also facilitates the incorporation of analytical tools and data in order to faithfully imitate real-world behavior. Engineers can check the structural performance and integrity of their designs by connecting their bridge models to structural analysis tools. By utilizing an integrated approach, the design process is more confidently assured that the digital representation closely resembles the actual reality.

Revit also provides tools for cooperation and clash detection among diverse disciplines. This is particularly important for bridge construction since a number of different systems, including roads, utilities, and drainage, must be coordinated with the bridge structure. Conflict detection assists in spotting possible issues early in the design process, reducing the need for expensive rework during construction.

Finally, visualization tools are crucial for communicating design purposes and enhancing stakeholder interaction. Bridge designers may produce realistic simulations that depict how the bridge will look in its ultimate location using the rendering tools in Revit. This promotes a greater comprehension of the project's aesthetic and functional features and helps in securing approvals, presenting ideas to non-technical stakeholders, and getting things done.

To sum up, bridge modeling methods in Revit provide designers and engineers with a full range of tools that address every facet of bridge design, analysis, coordination, and communication. Professionals may develop complex bridge models that faithfully depict real-world structures and contribute to the effective execution of safe and creative bridge projects by utilizing parametric modeling, adaptive components, analytical integration, collision detection, and visualization.

3.2. Visual programming in bridge modelling

3.2.1. The Pros and Cons of Visual Programming

An entirely new paradigm in the creation and management of intricate architectural models and systems is introduced by visual programming in Revit. The accessibility of visual programming to non-programmers is a key benefit. For many design professionals, traditional coding frequently necessitates a thorough understanding of programming languages. Contrarily, node-based visual programming interfaces let users connect predefined blocks or nodes to construct complicated logic devoid of the need for programming. This stimulates cross-disciplinary cooperation and simplifies the design process.

Visual programming in Revit can have drawbacks, too. Learning a new interface and process can be challenging, which could be a negative. Even though visual programming is intended to be user-friendly, it still takes time to become adept. A more sophisticated comprehension of programming ideas may also be required if users are asked to develop custom nodes due to complex programming logic. Energy simulation and Lifecycle Analysis (LCA) are integrated plugins in BIM software like Revit.

The possibility for model complexity and performance problems is another factor to take into account. Visual programming can result in complex networks of interconnected nodes that are challenging to manage, which could have an effect on model performance and make long-term model maintenance more difficult. To reduce these difficulties, careful planning and optimization are required.

Non-programmers find it simpler to comprehend and take part in the development process because to visual programming. The interface's visual design does away with the requirement for in-depth coding expertise, making software creation accessible to a wider audience. The creation and manipulation of code elements is often made simpler for developers by the user-friendly interface seen in most visual programming environments. Programming can be done more logically and intuitively thanks to the graphical representation of code. Developers may easily prototype and iterate their ideas thanks to visual programming. Without having to write and troubleshoot numerous lines of code, it is simpler to explore, test, and fine-tune concepts when code elements are represented visually.

Software development is not complete without debugging, and visual programming makes debugging easier. Visual representations of code elements make it simpler to spot and correct errors, which cuts down on the amount of time spent debugging. Multiple developers can work on the same project at once thanks to the collaborative capabilities that are frequently supported by visual programming environments. Real-time cooperation is made possible and teamwork is encouraged. Data flow is frequently represented visually in visual programming, which makes it simpler to understand and control. Developers can more easily understand the logic of their code when they can see how data flows between and transforms between various parts. Certain sorts of programming errors are frequently avoided by built-in validation techniques in visual programming environments. These measures can aid in the early detection of mistakes, leading to more robust and dependable code.

By enabling developers to construct unique functions or modules and reuse them across several projects, visual programming encourages code reuse. By doing this, development time and effort can be greatly reduced. Visual programming relies less on complicated syntax than conventional programming languages, which lowers the risk of syntax errors. Beginners will find it simpler to get started as a result, and debugging will take less time. When creating complicated systems with numerous components and interactions, visual programming is especially helpful. The complexity of such systems is simpler to comprehend and handle thanks to the graphical representation.Flowcharts, which are frequently employed in many different industries, resemble visual programming frequently. Because of their experience, non-programmers find it simpler to understand the logic and organization of the code.

Compared to conventional text-based coding, visual programming often has a more rapid learning curve. The environment's graphical features make it easier to understand programming ideas and enable beginners to get started coding right away. By explaining programming principles to pupils in a more interesting and dynamic way, visual programming can be a potent educational tool. It enables children to grasp the basics of coding and reasoning without becoming daunted by syntax. Developers can simply add and arrange code pieces with the help of drag-and-drop capabilities, which is frequently provided by visual programming environments. This streamlines the coding procedure and lowers the possibility of errors.

With additional software development tools like version control systems and debugging tools, visual programming tools frequently work well together. This improves project management and the entire development experience. Developers can see the outcomes of their activities in real-time thanks to visual programming environments' instant feedback on the code. This interactive quality fosters experimentation and expedites the development process. In particular, visual programming is advantageous for creating user interfaces. It is simpler to build and adapt layouts when UI elements are represented graphically, which leads to interfaces that are more aesthetically pleasing and user-friendly.

Particularly for projects with intricate user interfaces or sophisticated logic, visual programming can greatly shorten the development process. Faster development cycles are a result of the environment's intuitiveness and the reuse of code components. It is simpler to represent complex data sets in a visual and interactive way since visual programming platforms frequently come with built-in support for data visualization. For scientific applications and data processing, this is especially helpful. Since many visual programming tools are cross-platform, programmers can create software that works on several operating systems without making major changes. This enhances the produced applications' portability and accessibility. The entry barrier for people who want to learn programming is lowered through visual programming.

Visual programming is advantageous for both developers and designers. It enables designers to swiftly test various design concepts and prototype ideas without depending on developers to put them into practice. Developers can see how data flows through a program by using the visual feedback that visual programming environments frequently offer on data flow. Understanding complex algorithms and improving code performance can both benefit from this. By visualizing the execution flow, visual programming makes it simpler to understand and debug complex code. Developers are better able to trace the execution path and see any problems or bottlenecks. Automatic documentation generation is a common feature of visual programming environments, giving valuable information on the logic and structure of the code. In the long run, this can be useful for maintaining and updating software.

The development of intelligent, interconnected systems is facilitated by the easier connection and operation of various IoT devices due to the graphical representation of code elements. For people with certain difficulties like dyslexia or visual impairments, visual programming may be more accessible. Alternative ways to interact with the code are provided via the interface's graphical design. Projects involving machine learning may benefit from using visual programming. Machine learning algorithms' outputs can be visually inspected and analyzed by developers, which helps them comprehend and enhance their models. A useful tool for communicating with non-technical clients or stakeholders is visual programming. Real-time collaboration is frequently supported by visual programming environments, making it simpler for remote teams to work on a project simultaneously. This enhances coordination and communication, resulting in quicker development cycles.

It is possible for visual programming environments to place some restrictions on what can be done. The freedom of developers may be limited because the specified collection of graphical elements and their functionalities may not cover all conceivable programming scenarios.Beginners may find it easier to understand visual programming, but understanding more complex ideas and methods might still be difficult. It could take more work to comprehend the underlying logic and structure of complicated applications. The expressiveness of visual programming environments could not be as high as that of conventional text-based coding. Some difficult programming concepts could be more difficult to visualize, resulting in a loss of accuracy or clarity. When working on large and complicated projects, visual programming may experience scalability issues. Performance and maintainability can suffer from managing a high number of connections and code pieces.

In comparison to conventional programming languages, the community for visual programming tools may be smaller. As a result, developers may have fewer tools, tutorials, and libraries at their disposal, which can make it more difficult to locate assistance or fixes for particular issues. The portability of code may be hampered by the frequent use of proprietary frameworks or formats in visual programming environments. It may be difficult or perhaps necessary to rewrite some of the code when moving projects between platforms or tools. Visual debugging might be helpful, but it can also become more difficult for some issues. It could take more work to comprehend how various code pieces interact with one another and determine the true source of an issue. The representation of sophisticated algorithms or abstract notions that are difficult to visualize may be a challenge for visual programming.

Due to the additional levels of abstraction and required interpretation, visual programming environments may cause performance overhead. For computationally intensive applications in particular, this may affect how quickly the produced software executes. Specific tools or software platforms are frequently required for the proper operation of visual programming. Because of this dependence, developers may have fewer options, be more tied to one ecosystem, have less flexibility, and may even be forced to use a certain technology stack.

Furthermore, the study of coding-based solutions could be constrained by the reliance on visual programming. Despite the fact that visual programming is fantastic for many design jobs, there may be circumstances in which traditional coding provides more exact control or specific functionality that cannot be easily delivered with node-based methods.

In conclusion, the benefits and drawbacks of visual programming in Revit should be compared depending on the particular requirements and objectives of a project and the expertise of the design team. Its ease of use, adaptability, and capacity for quick experimentation can significantly improve the design process. When considering whether to use this method, one should take into account the learning curve, potential model complexity, and restrictions compared to traditional coding.

3.2.2. Guidelines for using visual programming in Bridge modeling

Modern CAD (Computer Aided Design) software is being gradually replaced by BIM (Building Information Modelling). Its goal is to replace disparate building data dispersed across several files with a single unified model incorporating all building data (Salamak et al., 2019). In Autodesk Revit, visual programming has become a potent tool for modeling bridges, giving engineers and designers more freedom, productivity, and accuracy. To properly utilize the advantages of visual programming in your bridge modeling process, abide by the following rules:

1. Have a Basic Understanding of Visual Programming: Learn the core terms used in Revit's visual programming language, such as nodes, connectors, and data flow. This knowledge serves as the basis for building intricate bridge models.

2. Select the Appropriate Visual Programming Platform: Dynamo is one of many visual programming platforms that Revit supports. Choose the platform that best fits the demands of your project and your skill set.

3. Outline Your Workflow: Before beginning visual programming, plan your workflow for bridge modeling. Determine the important processes, components, results, and potential difficulties. Your ability to create a structured and effective visual program is aided by this planning.

4. Dissect Complicated jobs: Break down complex bridge modeling jobs into more simpler, more compact tasks. To make it simpler to maintain and troubleshoot your models, create separate visual programs for each of these subtasks.

5. Give Data Management a Priority: Good data management is essential. Use the right data structures to efficiently organize the information. To guarantee correctness throughout the model, keep your data naming conventions and unit definitions consistent.

6. Make Use of Pre-Built Nodes: Dynamo and other visual programming environments include pre-built nodes for common tasks. These nodes speed up and shorten the modeling process. However, make sure that the criteria for your project are met by these nodes.

7. Customization for Particular Bridge Types: There are many different types and designs of bridges. Adapt your visual programs to the particular requirements and geometrical limitations of the bridge type you're working on.

8. Include parametric design: Use visual programming to harness the potential of parametric design. Construct models that adapt quickly to changes in the input parameters to facilitate effective design exploration.

9. Consistent Testing and Refinement: Before using your visual programs on challenging bridge projects, continuously test them on simplified models. Iteratively improve your programs to guarantee precision, dependability, and top performance.

To sum up, visual programming is an effective method for modeling contemporary bridges in Revit. By adhering to these rules, you may make the most of your skills to rapidly construct complex bridge models, promote teamwork, and produce excellent engineering projects.

3.3. Automatic or semiautomatic generation of models based on visual programming

The ability to automatically or semi-automatically create complicated models with improved efficiency and accuracy has altered the way models are developed in several industries. This strategy has enormous potential for fields like manufacturing and architecture. Visual programming-based automatic or semi-automatic model generation greatly expedites the design process. Models can be generated with little manual input by specifying a set of rules and parameters, freeing designers to concentrate on enhancing concepts and investigating alternative designs.

By adhering to set guidelines, visual programming promotes correctness and consistency. By doing so, human input errors are avoided, resulting in more accurate models and a lower likelihood of design defects. The development of parametric models that can adjust to changing input parameters is made possible by visual programming. With the help of this function, designers can easily generate and evaluate many design choices. Although automatic generation is efficient, semiautomatic methods give more customization and control. At pivotal points, designers can step in to fine-tune the model in accordance with the requirements of a given project and their own design goals. Although automatic generation is efficient, semiautomatic methods give more customization and control. At pivotal points, designers can step in to fine-tune the model in accordance with the requirements of a given project and their own design goals. When designing architectural big span buildings or footbridges, architects and engineers must work closely together to co-create aesthetically pleasing and structurally sound shapes that suit the goals of utility, economy, and aesthetics. An immediate response is frequently anticipated during the co-creation process (Chong et al,.2020).

By automating repetitive activities, visual programming tools like Dynamo enable architects, engineers, and designers to streamline their workflows. Faster model development and change results from this. Parametric modeling, which enables models to alter dynamically based on changing parameters, is made possible through visual programming. This adaptability encourages the investigation of many design possibilities and the effective iteration of such designs. By following predetermined standards and guidelines, automatic model generation contributes to maintaining design consistency. This is especially important for big projects when following rules is key. Human error is less likely to occur in automated processes. Visual programming reduces the possibility of errors during model construction, eliminating the need for labor-intensive manual adjustments. Designers are able to swiftly prototype concepts, test ideas, and see several design options with semiautomatic or automatic generation. In the first stages of design, this quickens the decision-making process (Hartung et al,. 2020).

Engineers and architects can write custom scripts using visual programming to meet the needs of certain projects. This flexibility is especially useful for projects that provide particular design issues. Visual programming tools can be used to produce complex geometries with precision that are difficult to create manually. This is very helpful in architectural designs that incorporate complex forms. Model generation that is automatic or semi-automatic can easily be integrated with analysis tools. By connecting created models to simulation software, engineers may swiftly assess structural, energy, or environmental performance. Collaboration is improved by automated or semiautomated models because they give a clear visual representation of design intent. This makes it easier for interdisciplinary teams to communicate effectively.

The use of BIM technology can be extended to the planning, assessment, design, construction, operation, and maintenance of transportation infrastructure, as well as the realization of data sharing among all project participants over the course of the project's entire life cycle. It can also be used to support fine design, industrial construction, and industrial chain connections, as well as the improvement of engineering environment and energy efficiency. (Jinjin, 2023)

The project's analysis, inspection, and simulation can serve as the basis for the scheme optimization and scientific decision-making throughout the entire project process; they can also support the collaborative work of various disciplines and foster conditions for the transportation industry's improvement in quality and efficiency, energy conservation, and environmental protection.While employing visual programming tools might make some elements of model generation simpler, there may be a learning curve involved in doing so. Teams should set aside time for skill building and training. To ensure accuracy and compliance with project standards, automated models should go through extensive quality control assessments. The integrity of the created models must be regularly maintained through audits and validations. Even though automating procedures can increase productivity, it's important to keep the creative and imaginative components of design. Within automated workflows, designers should still be able to use their creative judgment.

In conclusion, the automatic or semiautomatic production of models using Revit's visual programming tools has a revolutionary impact on the BIM and design industries. Professionals that use these tools well can produce models more quickly, more correctly, and with greater flexibility, which improves project outcomes and elevates design quality.

This page is intentionally left blank

4. CASE STUDY

4.1. Description of selected bridge design case

The Brembo river, which separates the town into two sections, is where Ponte San Pietro is located. It is regarded as the first town coming from Bergamo in the region known as Isola, which is located about 7 kilometers west of the orobic capital and is bordered by the waters of the two main rivers, Adda and Brembo, as well as by the distinct division of valleys and orobic mountains in front. Isola is a geographical area made up of 21 municipalities. The municipality is bordered to the north by Brembate di Sopra and Valbrembo, to the south by Presezzo and Bonate Sopra, to the west by Mapello and Presezzo once again, and to the east by Curno and Mozzo. the following, two pictures of the top and bottom views are given in numbers 4.1 and 4.2 (www.comune.pontesanpietro.bg.it, n.d.).

Figure 4.1 – San Pietro Bridge (Top)

Figure 4.2 – San Pietro Bridge (Bottom)

4.1.1. Old drawing sheets and their families in Revit

The bridge has 13 spans, 12 piers, two abutments, 12 foundations, and deck made of five concrete girder beams. The average length of beam and pier is respectively 35 and 10 meters. The drawing sheets are from Alpina company. the following, a picture of the bridge elevation is given in figure number 4.3

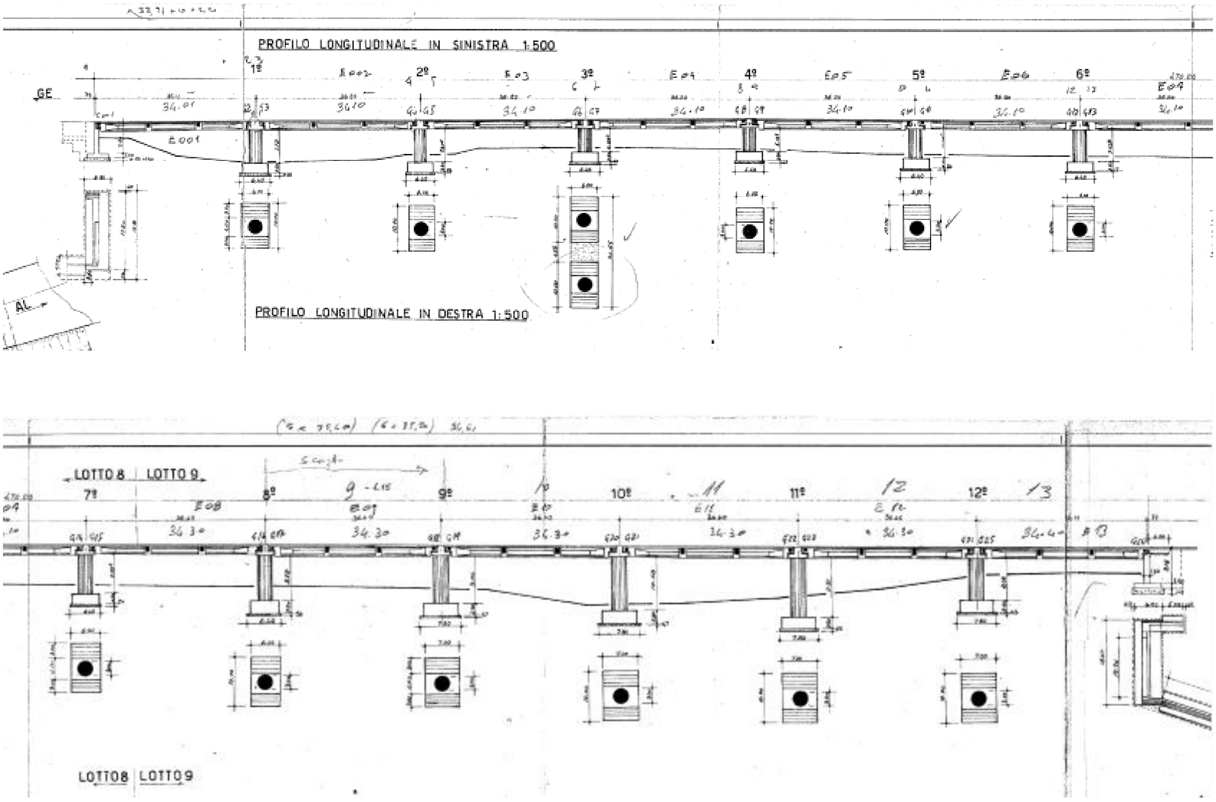


Figure 4.3 – The bridge elevation

There are two different types of foundations in this bridge. Designing a parametrically adaptable component that is simple to adjust for diverse project requirements is required when creating a foundation family in Revit. Launch Revit and open the family template editor to get started. Select the proper foundation category, such as "Structural Foundations," and then begin drawing the fundamental geometry of your foundation element, whether it be a spread footing, isolated footing, or slab. Define the foundation's dimensions and shape using the drawing and modification tools. The family should then include parametric controls to increase flexibility. Make use of parameters to control characteristics such as width, length, depth, reinforcement, and more. This enables users to modify the family's attributes and dimensions when adding it to a project. the following, four pictures of the foundations and their families are given in numbers 4.4 to 4.7.

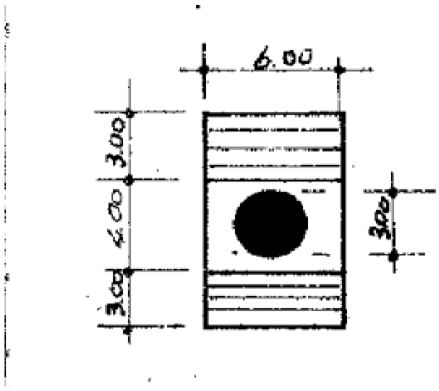


Figure 4.4 – Foundation(6x10)

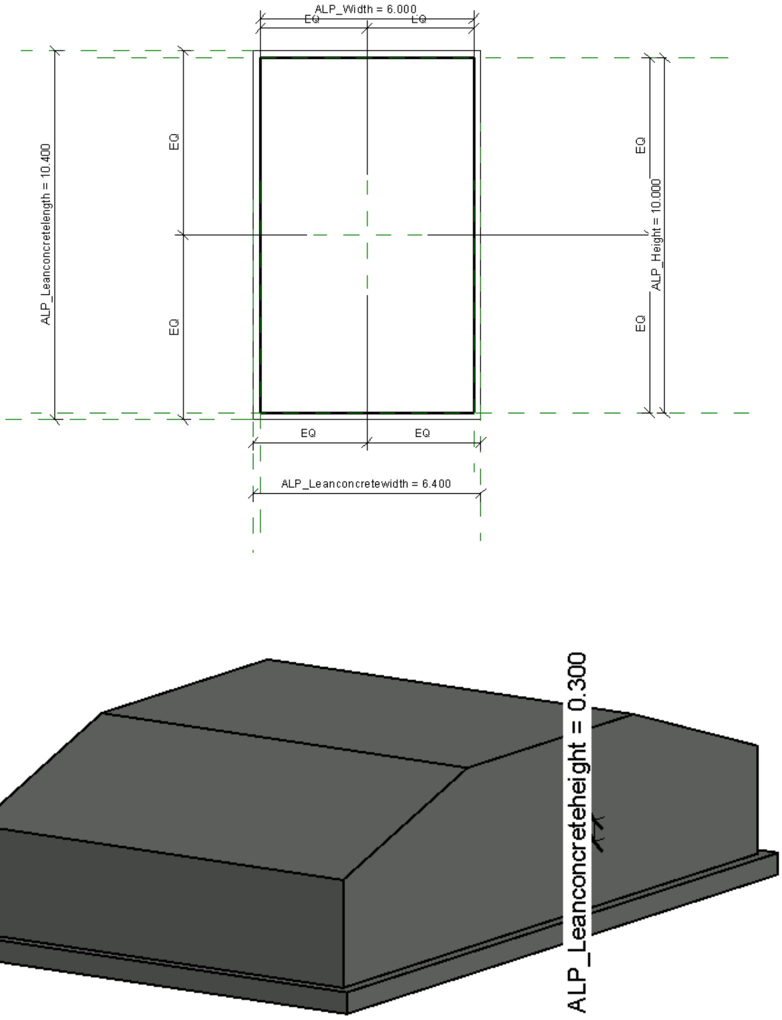


Figure 4.5 –Family of foundation (6x10)

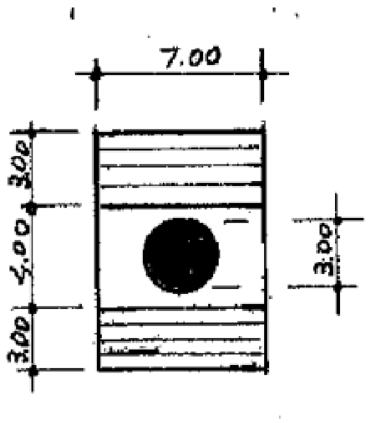


Figure 4.6 – Foundation(7x10)

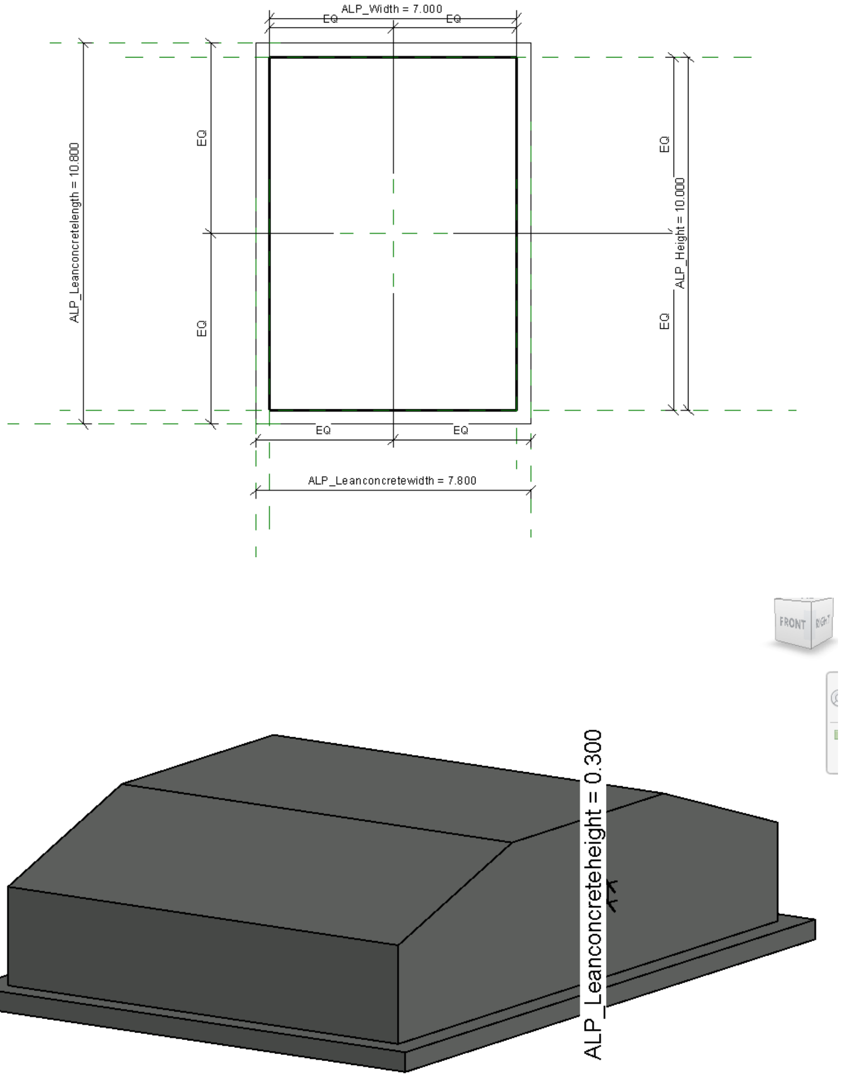


Figure 4.7 –Family of foundation (6x10)

There is one type of pier in this bridge. Open the software and proceed to the "Family" category to start a new family creation in order to establish a column family in Revit. Enter the family editor after selecting the template that best matches the architectural or structural type of the column. Create a 2D profile of the column using the editor's tools, and then produce a 3D representation of it by utilizing the extrusion or revolve methods. Set up flexible parameters for sizes, materials, and other attributes to enable for placement flexibility in projects. Add base and capital pieces through extra sketches and extrusions to improve detailing. To guarantee proper behavior, test the family's functionality in a sample project. the following, two pictures of the pier and its families are given in figures numbers 4.8 and 4.9.

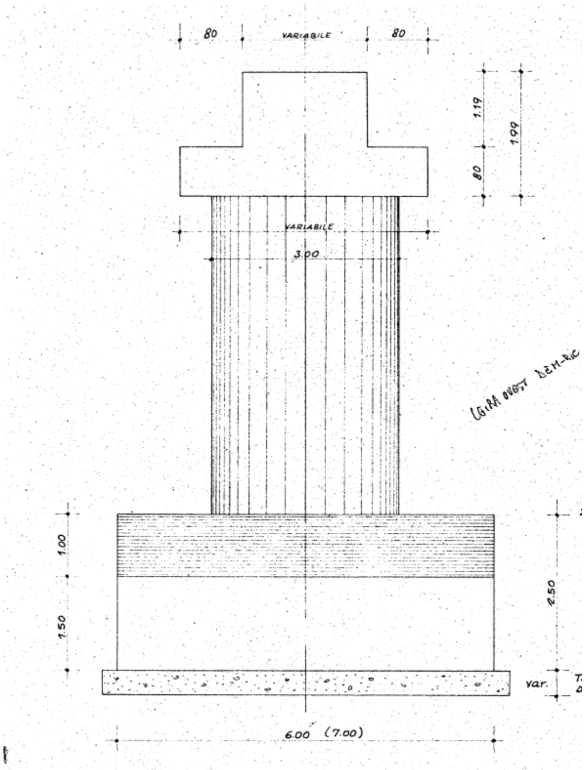


Figure 4.8 – Pier

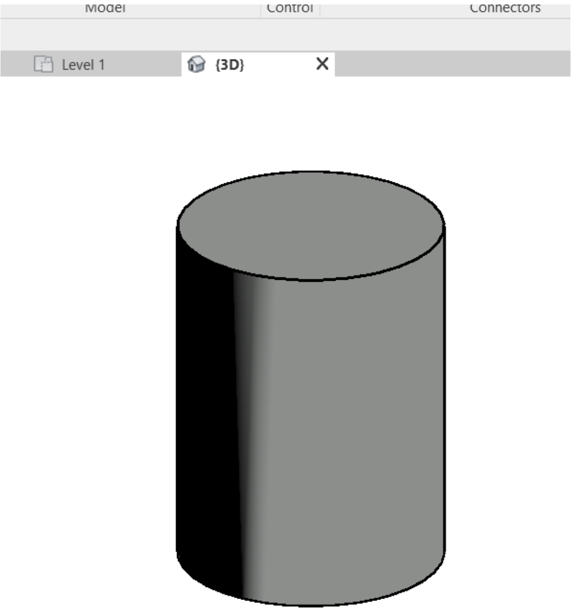
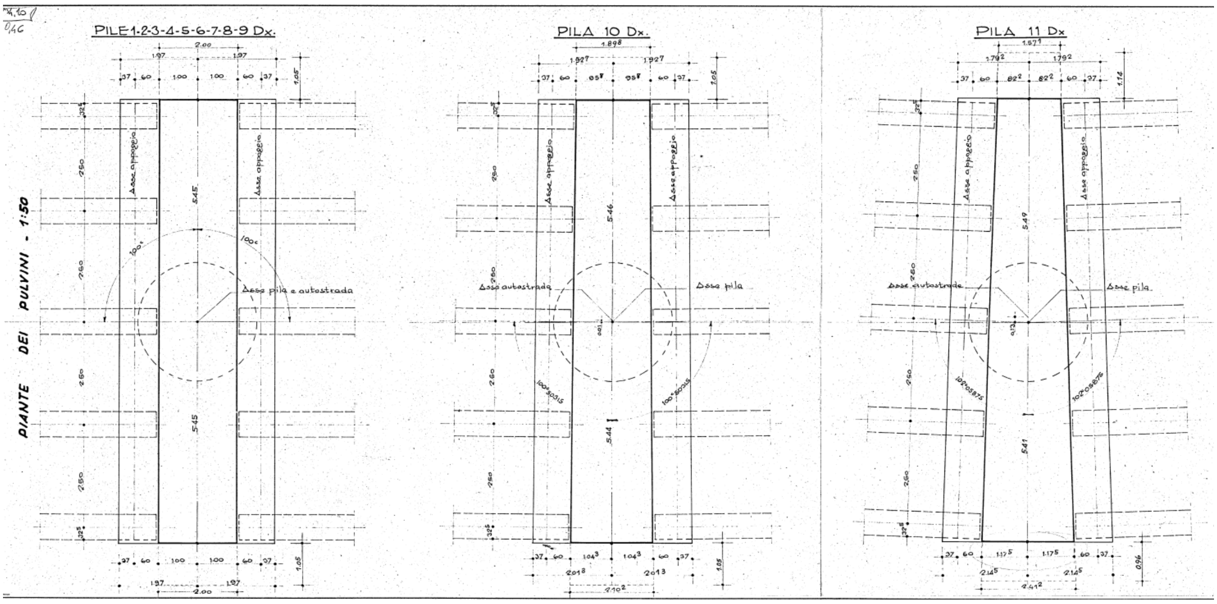


Figure 4.9 –Family of Pier

There are five types of piercaps in this bridge. the following, two pictures of the pier cap and its family are given in numbers 4.10 and 4.11.



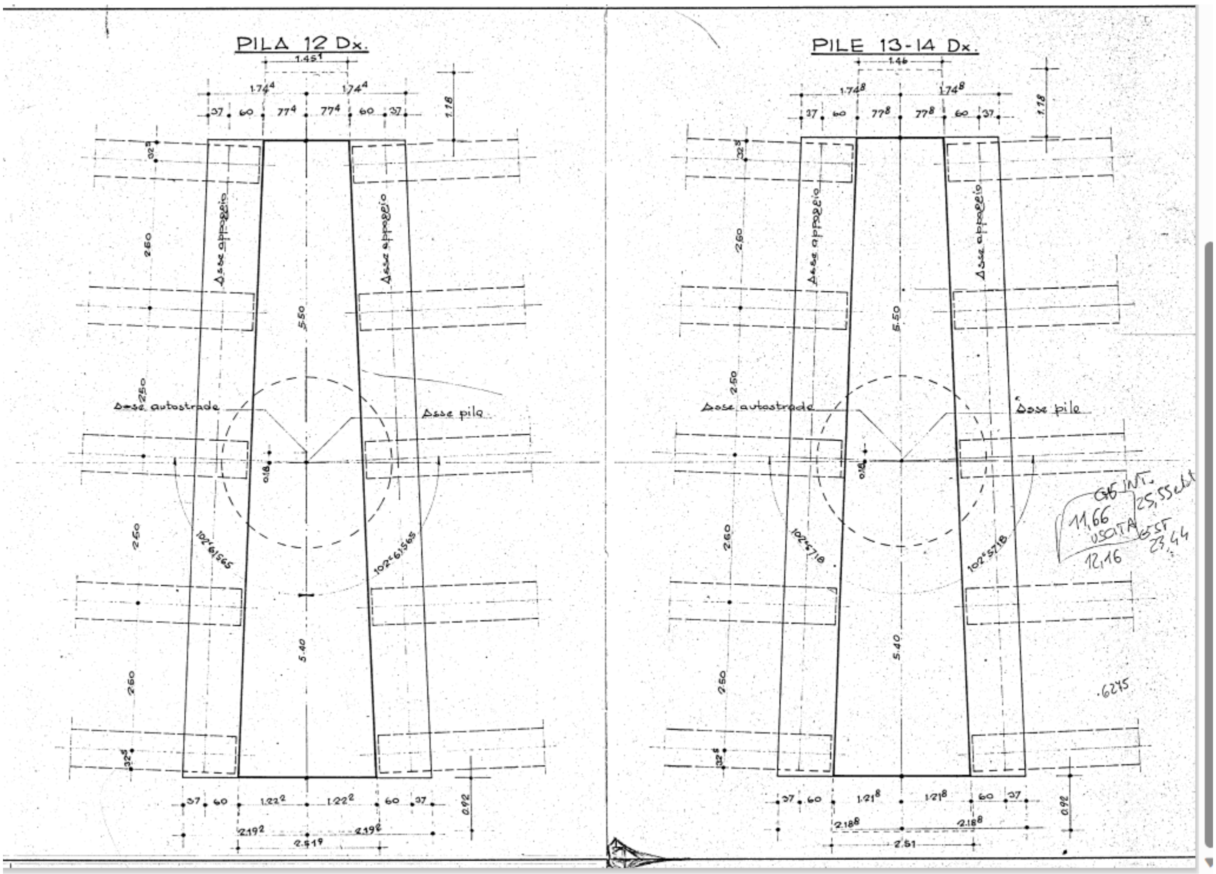
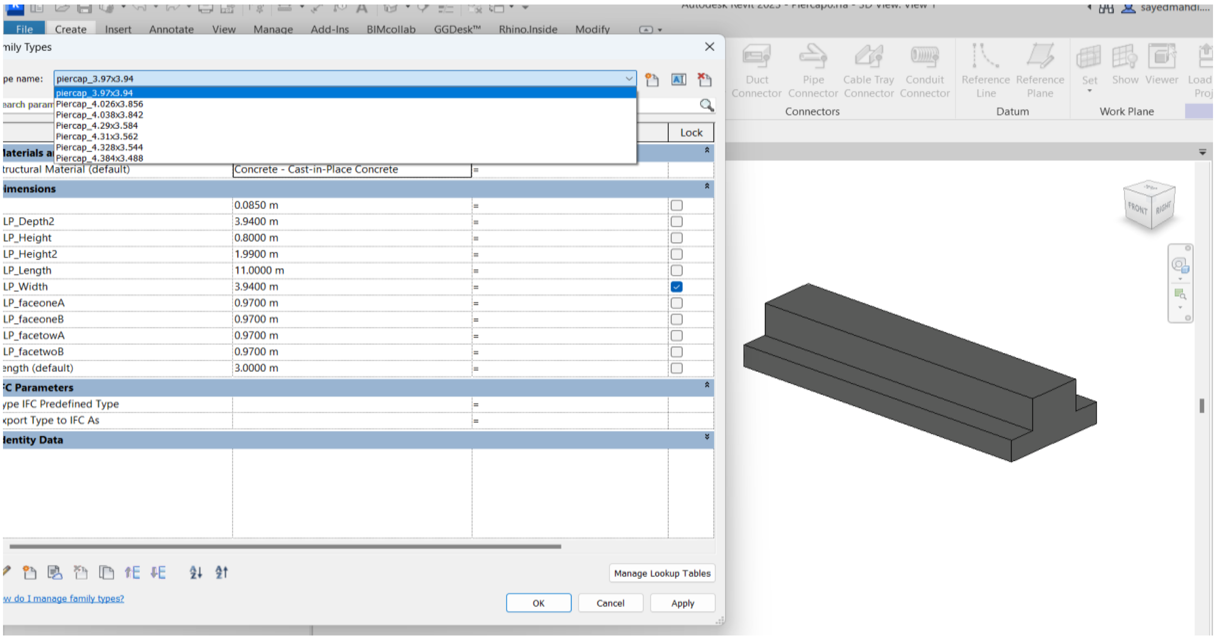


Figure 4.10 – Pier caps



mily Types

pe name: piercap_3.97x3.94

- piercap_3.97x3.94
- Piercap_4.026x3.856
- Piercap_4.038x3.842
- Piercap_4.29x3.584
- Piercap_4.31x3.562
- Piercap_4.328x3.544
- Piercap_4.384x3.488

earch param

laterials a			
tructural Material (default)	Concrete - Cast-in-Place Concrete	=	
imensions			
	0.0850 m	=	☐
LP_Depth2	3.9400 m	=	☐
LP_Height	0.8000 m	=	☐
LP_Height2	1.9900 m	=	☐
LP_Length	11.0000 m	=	☐
LP_Width	3.9400 m	=	☑
LP_faceoneA	0.9700 m	=	☐
LP_faceoneB	0.9700 m	=	☐
LP_facetowA	0.9700 m	=	☐
LP_facetwoB	0.9700 m	=	☐
ength (default)	3.0000 m	=	☐
C Parameters			
ype IFC Predefined Type		=	
xport Type to IFC As		=	
lentity Data			

Figure 4.11 –Families of Pier caps

The deck is made of five concrete girder beams and 20 cm concrete slab. The following actions are required to create a concrete beam family in Revit: Launch Revit first, then choose the "New" option under the "Family" category. Select an appropriate template, such as "Metric Generic Model" or "Imperial Generic Model." Draft the concrete beam's cross-sectional profile inside the family editor using the sketch tools. In order to turn the 2D profile into a 3D depiction of the beam, use the "Extrusion" command and modify the length as appropriate. Establish guidelines for measurements like width, height, and flange thickness to provide for flexibility in meeting different project requirements. Include extra parameters for characteristics like material requirements, reinforcing details, and visibility preferences. Include symbolic lines or annotation components, if desired, to represent internal reinforcement. Put the family into a practice to validate it. the following, two pictures of the concrete girder beam and its family are given in figures numbers 4.12 and 4.13. and two pictures of the concrete salb and its family are given in figures numbers 4.14 and 4.15.

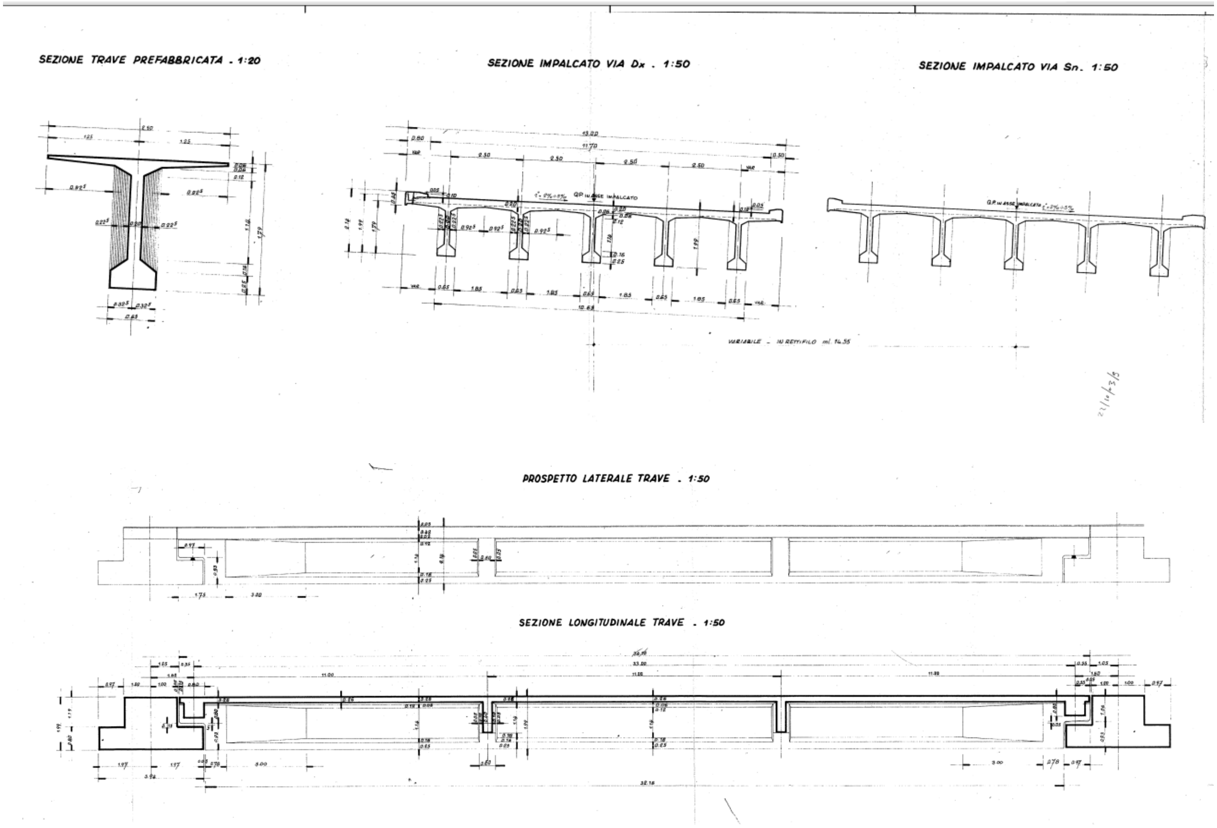


Figure 4.12 – Concrete girder beams

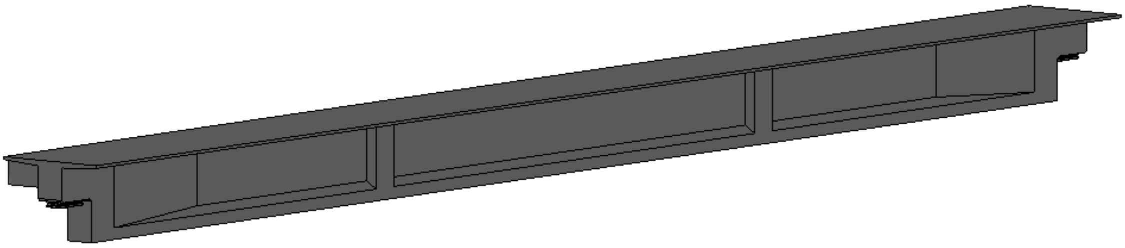


Figure 4.13 –Family of concrete girder beam

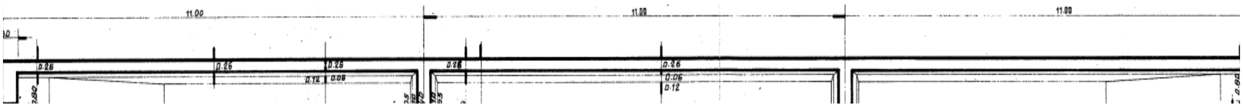


Figure 4.14 – Concrete slab

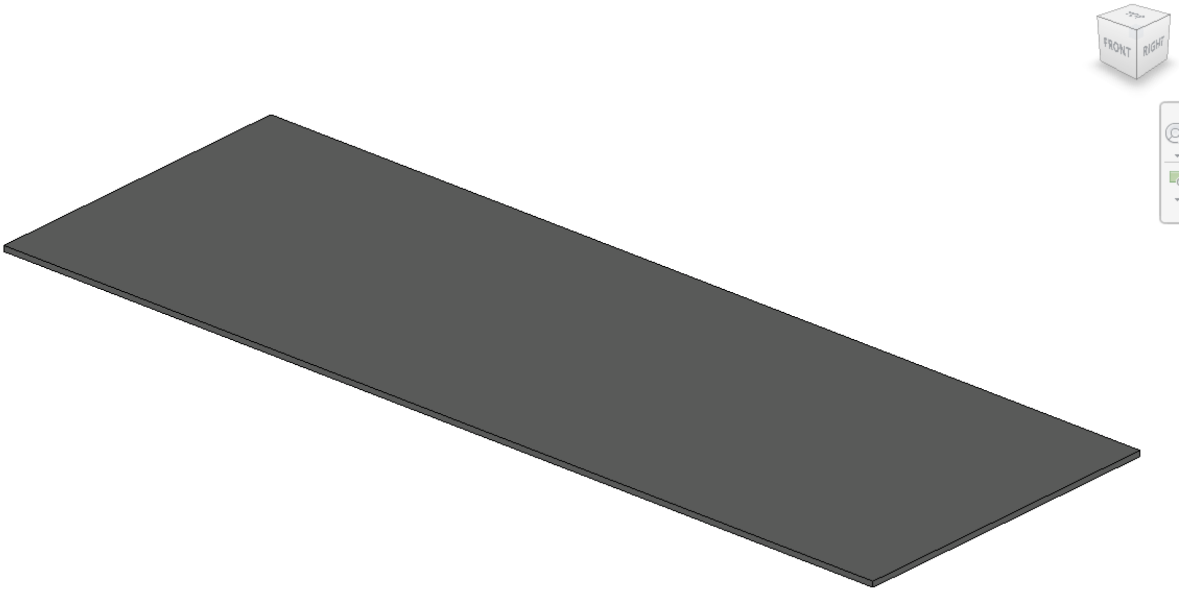


Figure 4.15 –Family of concrete slab

the following, two pictures of the guardrail and its family are given in figures numbers 4.16 and 4.17. and two pictures of the neoprene and its family are given in figures numbers 4.18 and 4.19.

Figure 4.16 – Guard rail

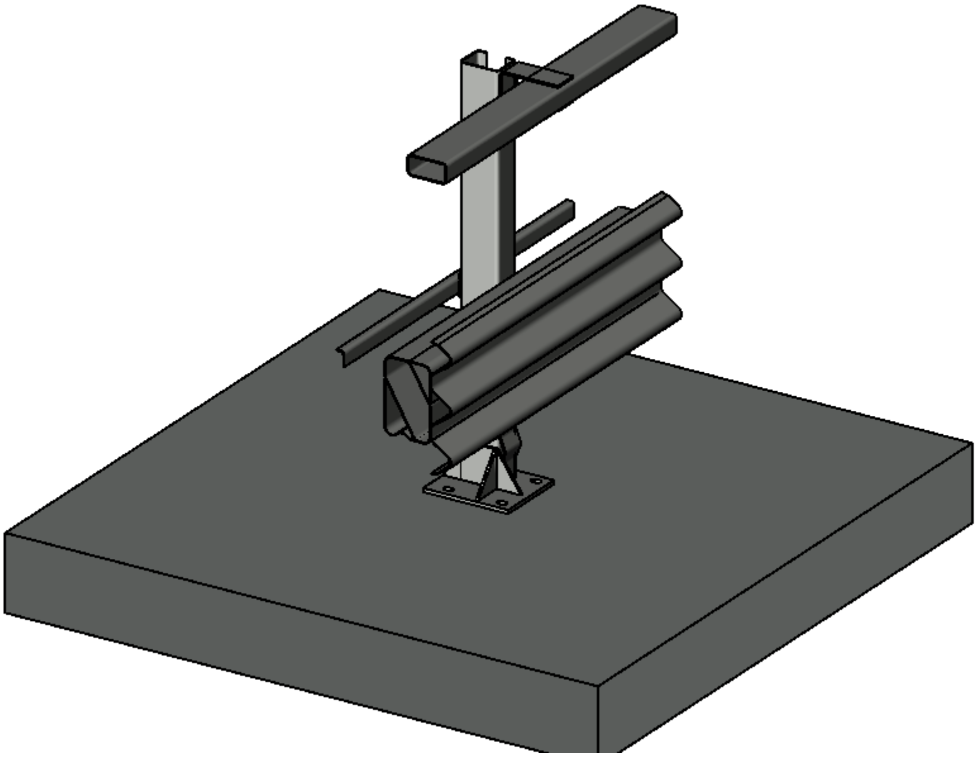


Figure 4.17 – Family of guard rail

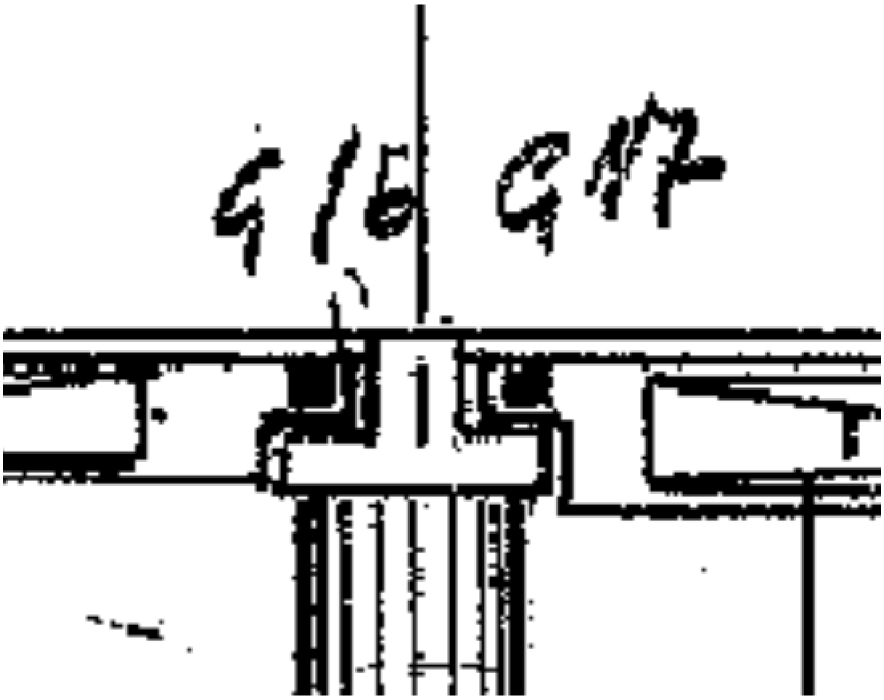


Figure 4.18 – Neoprene

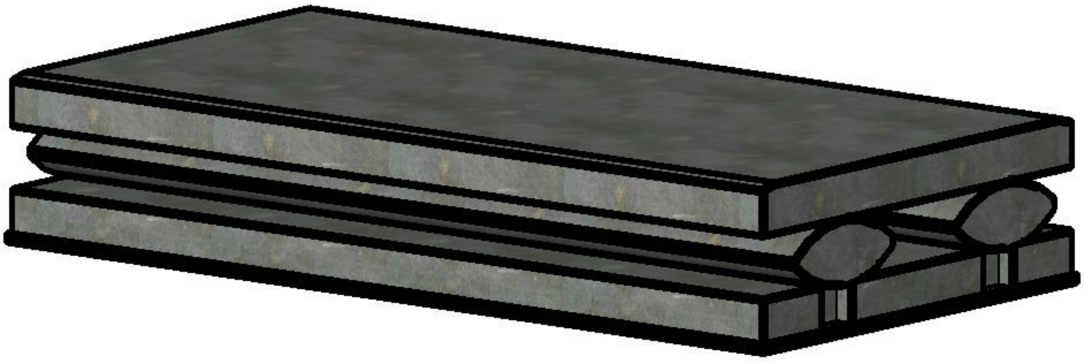


Figure 4.19 – Family of neoprene

the following, two pictures of the abutment and its family are given in figures numbers 4.20 and 4.21.

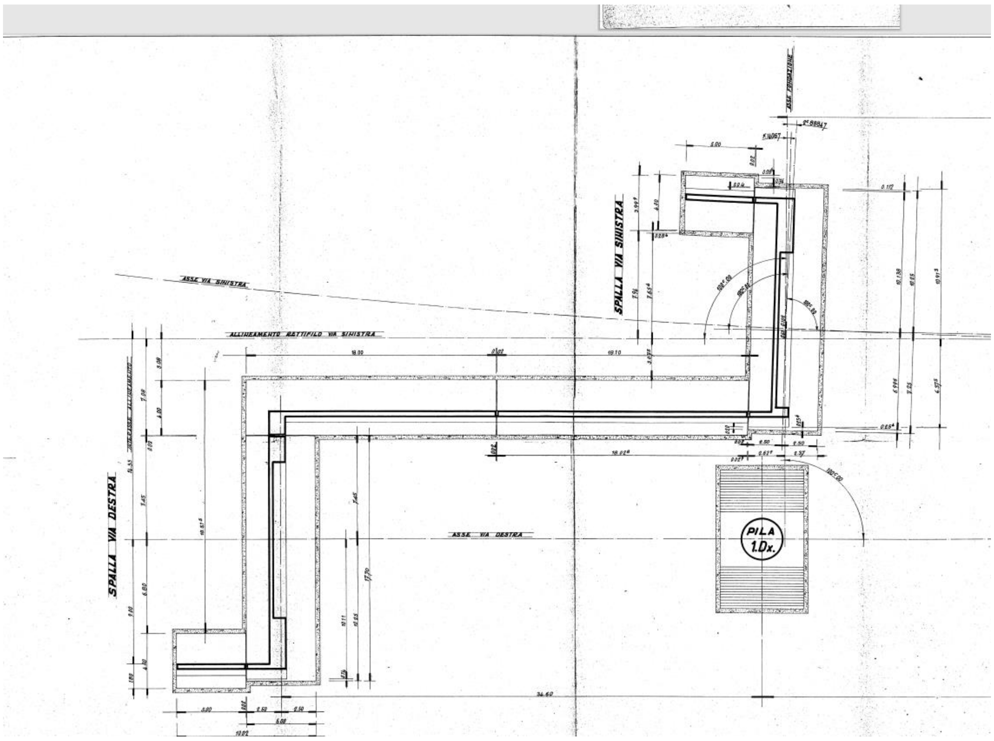


Figure 4.20 – Abutment

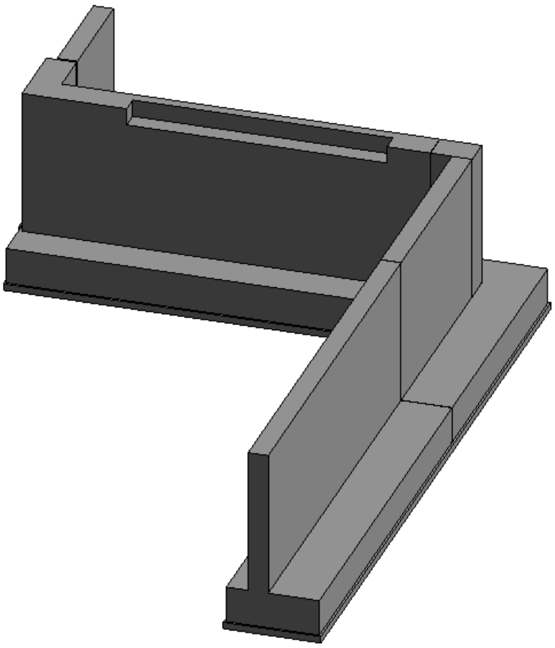


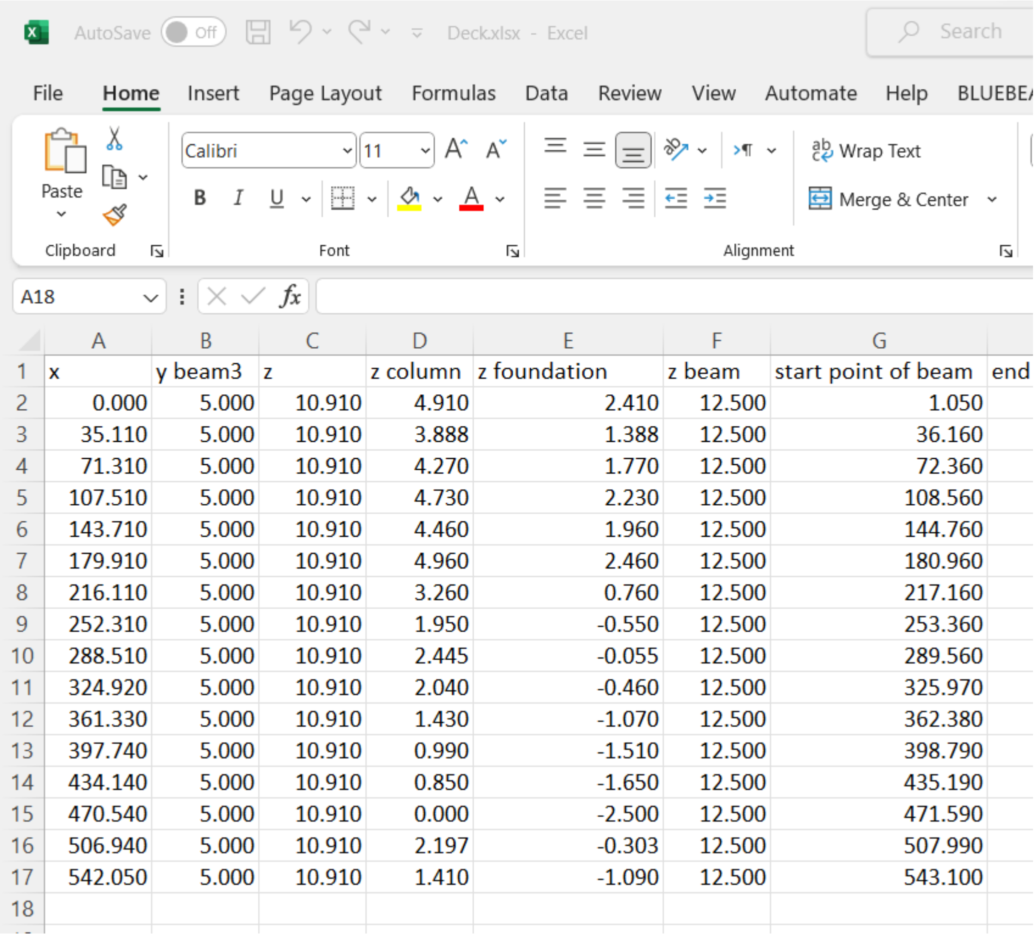
Figure 4.21 – Family of abutment

4.1.2. Modeling the entire bridge using points and families

In the network of the transportation system, bridges are essential. Dynamo involves a sequence of stages that use visual programming to automate and personalize the creation process when creating a structural element in Revit. Within the Revit environment, Dynamo is a potent tool that enables parametric design and automation (Shim et al., 2019).

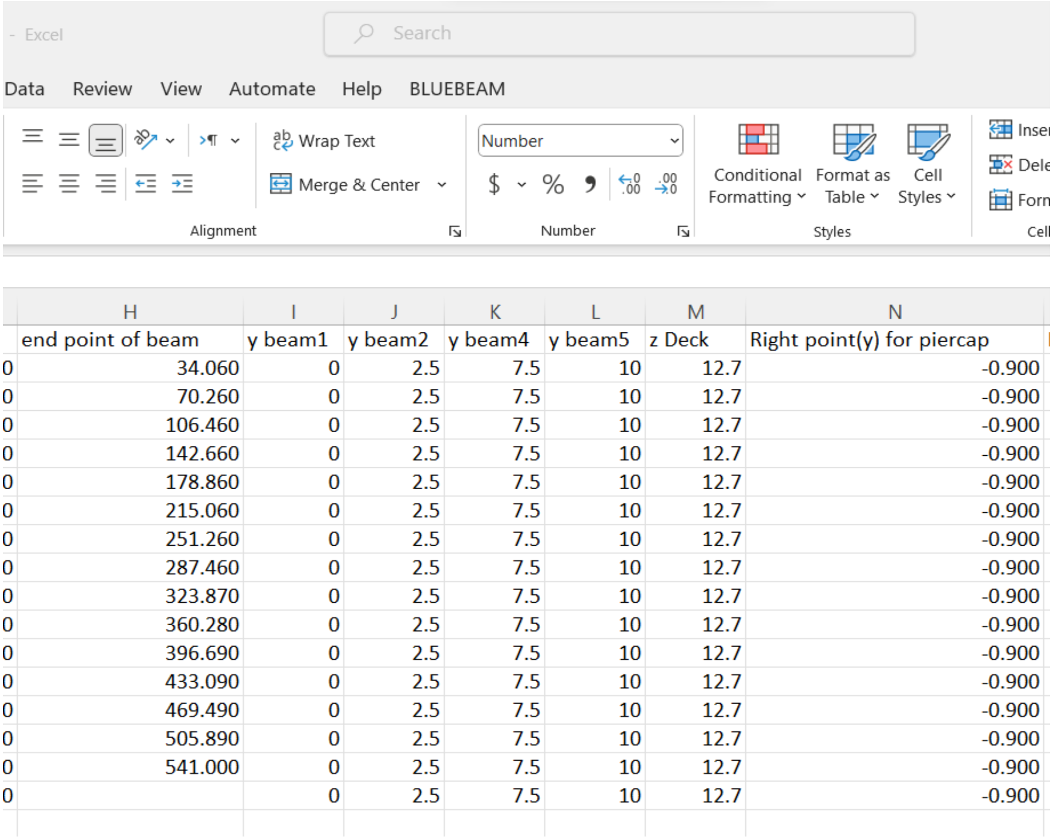
Open Dynamo from within Revit to get started. Make a new workspace for Dynamo. The "Revit. Elements" node must be used to establish a link between Dynamo and the Revit project. With the use of this node, Dynamo can communicate with native Revit components and features.

Establish the specifications needed for your structural element. Dimensions, levels, offsets, and other pertinent attributes may be included. Allow users to specify these parameters right in the Dynamo workspace by using input nodes. To speed up the procedure, you can also set default settings for certain parameters. In this research used points and lines to model the entire bridge. For points I used Excel. This Dynamo script can be used for new and existing bridges. The following, three pictures of the Excel are given in figures numbers 4.22 to 4.24.



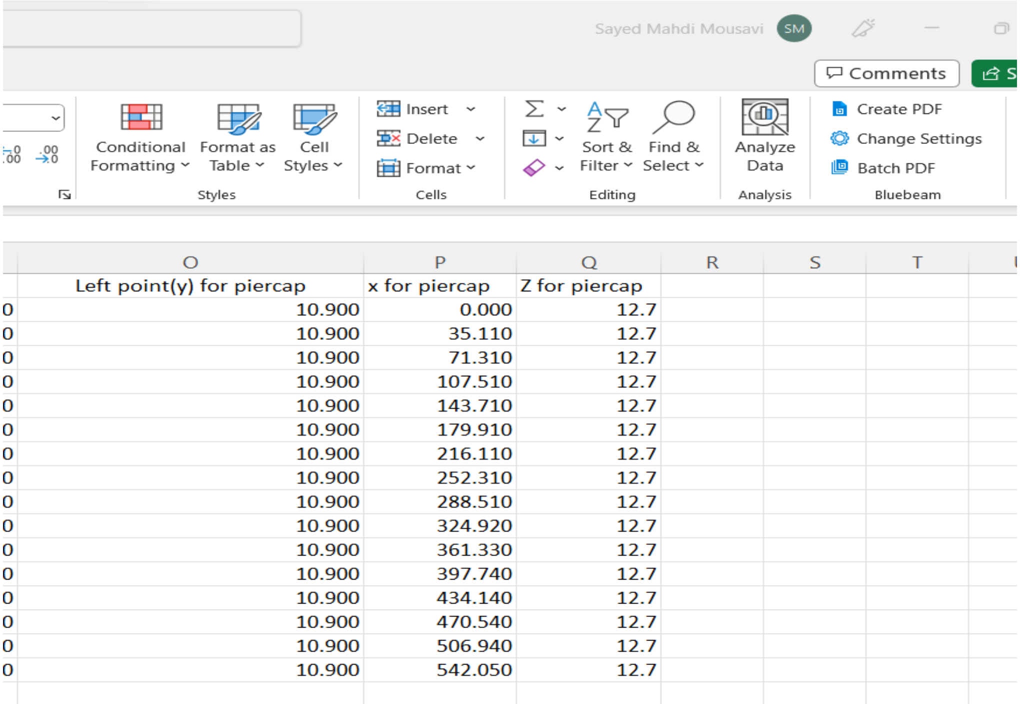
	A	B	C	D	E	F	G	
1	x	y beam3	z	z column	z foundation	z beam	start point of beam	end
2	0.000	5.000	10.910	4.910	2.410	12.500	1.050	
3	35.110	5.000	10.910	3.888	1.388	12.500	36.160	
4	71.310	5.000	10.910	4.270	1.770	12.500	72.360	
5	107.510	5.000	10.910	4.730	2.230	12.500	108.560	
6	143.710	5.000	10.910	4.460	1.960	12.500	144.760	
7	179.910	5.000	10.910	4.960	2.460	12.500	180.960	
8	216.110	5.000	10.910	3.260	0.760	12.500	217.160	
9	252.310	5.000	10.910	1.950	-0.550	12.500	253.360	
10	288.510	5.000	10.910	2.445	-0.055	12.500	289.560	
11	324.920	5.000	10.910	2.040	-0.460	12.500	325.970	
12	361.330	5.000	10.910	1.430	-1.070	12.500	362.380	
13	397.740	5.000	10.910	0.990	-1.510	12.500	398.790	
14	434.140	5.000	10.910	0.850	-1.650	12.500	435.190	
15	470.540	5.000	10.910	0.000	-2.500	12.500	471.590	
16	506.940	5.000	10.910	2.197	-0.303	12.500	507.990	
17	542.050	5.000	10.910	1.410	-1.090	12.500	543.100	
18								

Figure 4.22 – Points in Excel



	H	I	J	K	L	M	N
	end point of beam	y beam1	y beam2	y beam4	y beam5	z Deck	Right point(y) for piercap
0	34.060	0	2.5	7.5	10	12.7	-0.900
0	70.260	0	2.5	7.5	10	12.7	-0.900
0	106.460	0	2.5	7.5	10	12.7	-0.900
0	142.660	0	2.5	7.5	10	12.7	-0.900
0	178.860	0	2.5	7.5	10	12.7	-0.900
0	215.060	0	2.5	7.5	10	12.7	-0.900
0	251.260	0	2.5	7.5	10	12.7	-0.900
0	287.460	0	2.5	7.5	10	12.7	-0.900
0	323.870	0	2.5	7.5	10	12.7	-0.900
0	360.280	0	2.5	7.5	10	12.7	-0.900
0	396.690	0	2.5	7.5	10	12.7	-0.900
0	433.090	0	2.5	7.5	10	12.7	-0.900
0	469.490	0	2.5	7.5	10	12.7	-0.900
0	505.890	0	2.5	7.5	10	12.7	-0.900
0	541.000	0	2.5	7.5	10	12.7	-0.900
0		0	2.5	7.5	10	12.7	-0.900

Figure 4.23 – Points in Excel



	O	P	Q	R	S	T
	Left point(y) for piercap	x for piercap	Z for piercap			
0	10.900	0.000	12.7			
0	10.900	35.110	12.7			
0	10.900	71.310	12.7			
0	10.900	107.510	12.7			
0	10.900	143.710	12.7			
0	10.900	179.910	12.7			
0	10.900	216.110	12.7			
0	10.900	252.310	12.7			
0	10.900	288.510	12.7			
0	10.900	324.920	12.7			
0	10.900	361.330	12.7			
0	10.900	397.740	12.7			
0	10.900	434.140	12.7			
0	10.900	470.540	12.7			
0	10.900	506.940	12.7			
0	10.900	542.050	12.7			

Figure 4.24 – Points in Excel

The following Dynamo script models the entire bridge. They are shown in figures numbers 4.25 to 4.37.

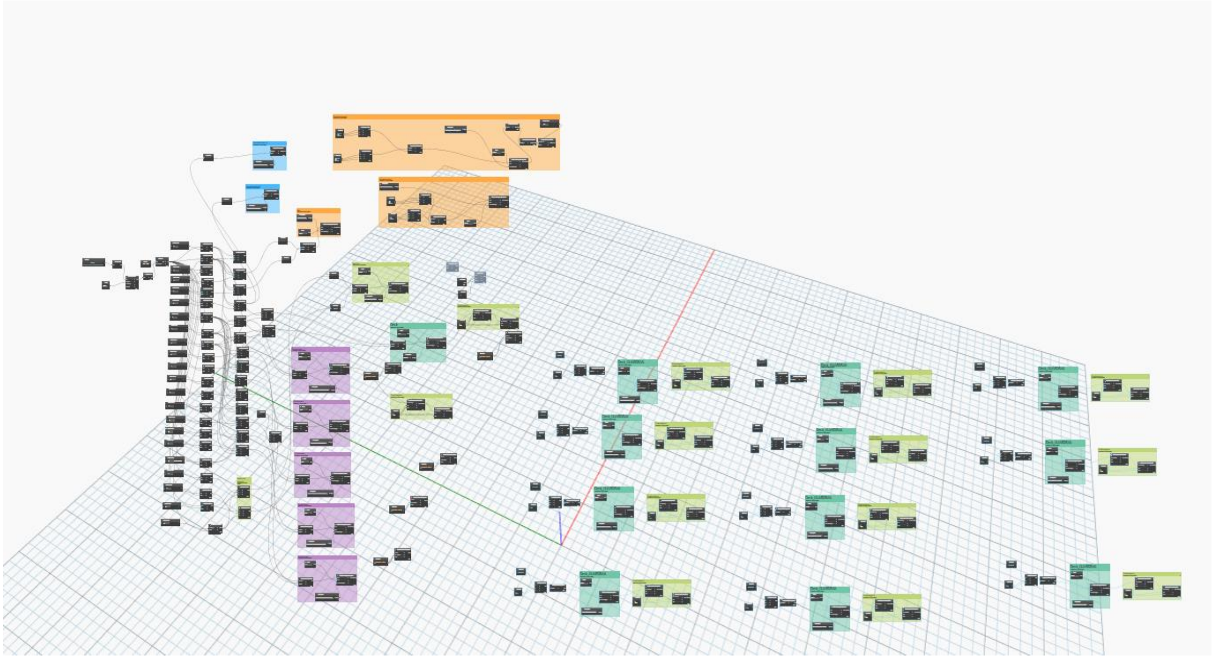
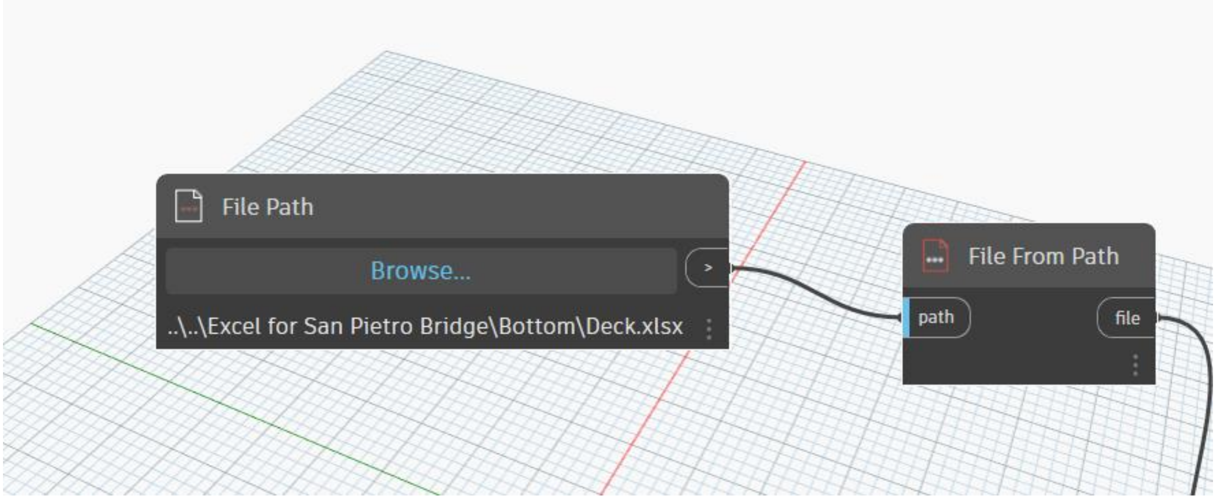


Figure 4.25 – Dynamo script

Follow these procedures to use a file path in Dynamo. Open Dynamo and launch a new script first. Next, import the required packages to give you access to file paths. The path to the file you want to utilize should then be saved in a file path variable that you establish within your script. Give the variable the proper value to specify the file path. Connect the file path to the script nodes or functions that require it next. The procedures involving the file path should then be carried out by running the Dynamo script. You may efficiently use a file path in Dynamo for your project by following these instructions. the following file path is shown in figure 4.26.



..\..\Excel for San Pietro Bridge\Bottom\Deck.xlsx

Figure 4.26 – File path

Open a new script in Dynamo to begin importing data from Excel. Afterward, import the programs required to work with Excel files. Connect the Excel file you want to import to the "Excel.ReadFromFile" node's "FilePath" input to specify the file path. Connect the "Excel.ReadFromFile" node's output to the "ExcelData" input to extract the data from the Excel file using the "Excel.ReadExcelData" node. Use different Dynamo nodes to manipulate the data, such as filtering, sorting, or doing computations. Use Dynamo's visualization nodes to visualize the data, or export it to different file types as necessary. Don't forget to save your Dynamo graph for alterations or future usage. the following file path is shown in figure 4.27.

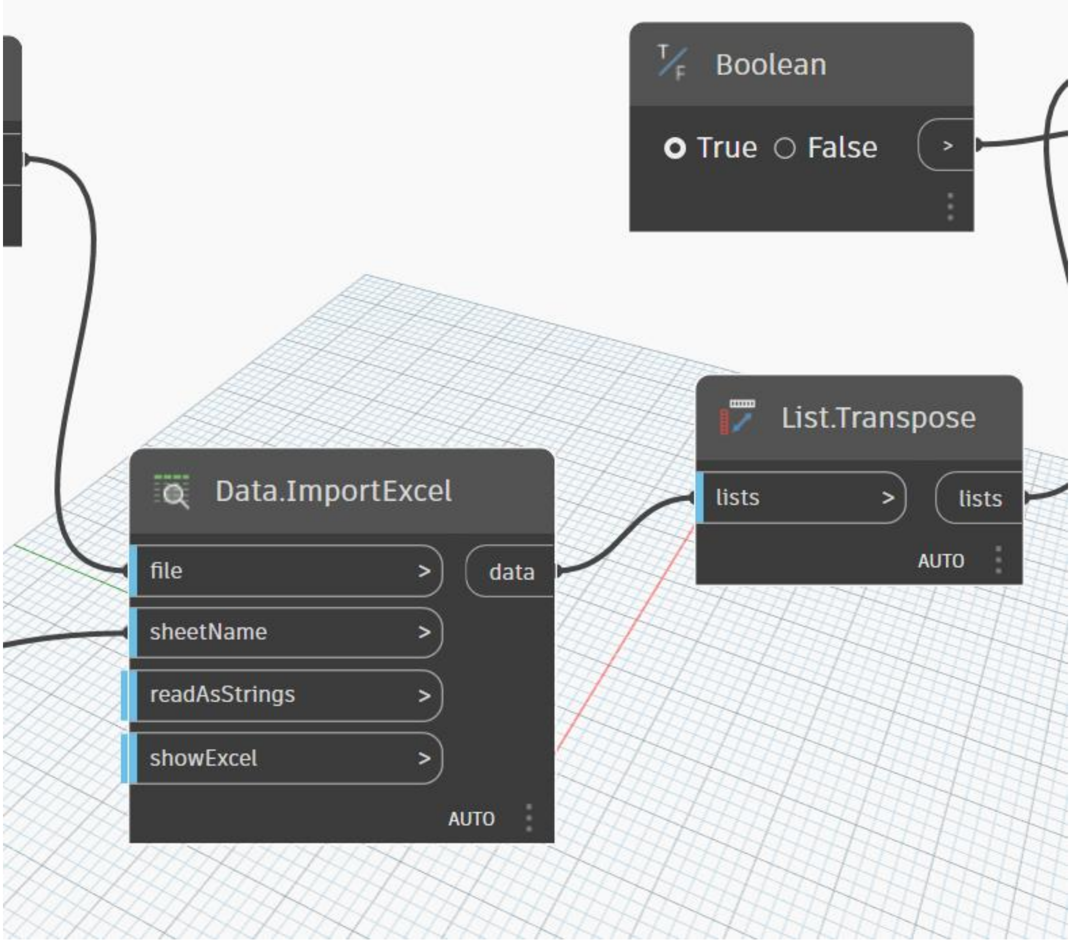


Figure 4.27 – Importing data from Excel

Use these instructions to use the List.Clean node in Dynamo. Create a new graph in Dynamo or open an existing one first. Then, import the required packages by choosing "Search for a Package" from the "Packages" menu. Enter "List.Clean" in the search box, then click on the matching package. The list you want to clear up should then be placed in a variable. To accomplish this, connect the "File Path" node from the "Core" library to the List's "FilePath" input by dragging it there.tidy node. Connect the List's output lastly.To the required output or more nodes in your network, create a clean node. the following List.Clean is shown in figure 4.28.

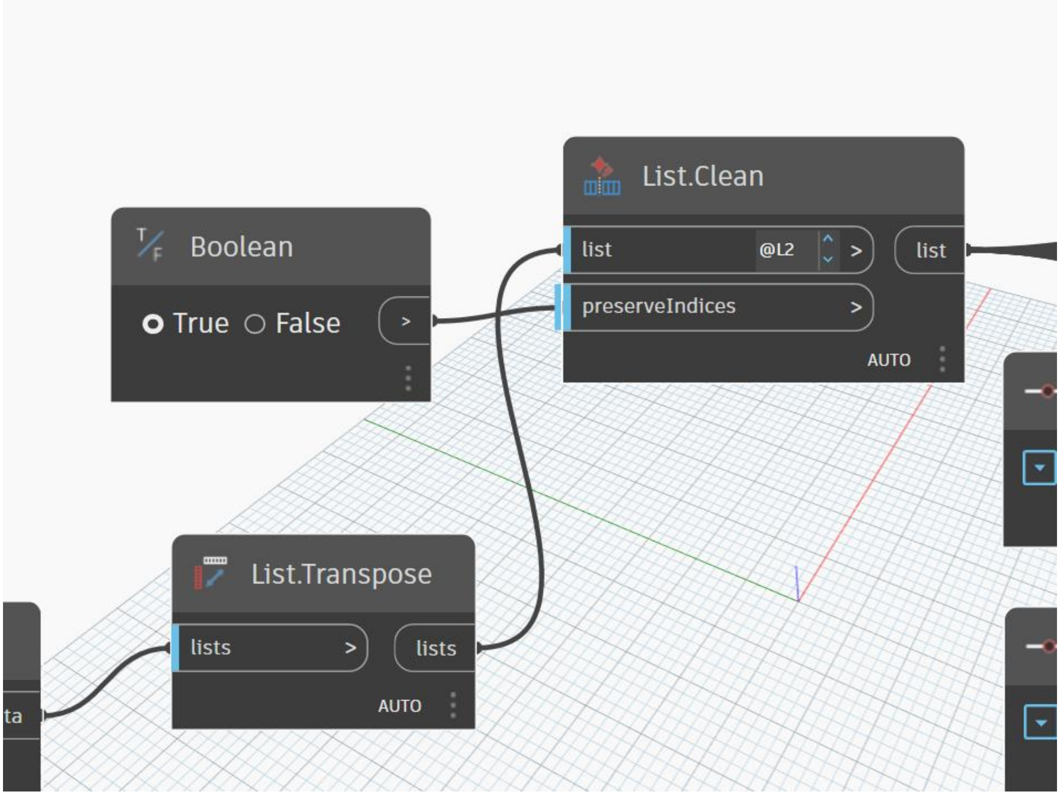


Figure 4.28 – List.Clean

You must first construct or open a graph in Dynamo in order to use a number slider. Then choose the "Number Slider" node by going to the "Input" tab. Put it on your graph canvas by dragging it there. To access the slider node's properties, double-click on it. You can configure the slider's minimum and maximum values as well as its step size in the properties pane. If necessary, you can also change the label and units. Once the slider has been set up, you can connect its output to other nodes in your network to dynamically control numerical values. the following Number Slider is shown in figure 4.29.

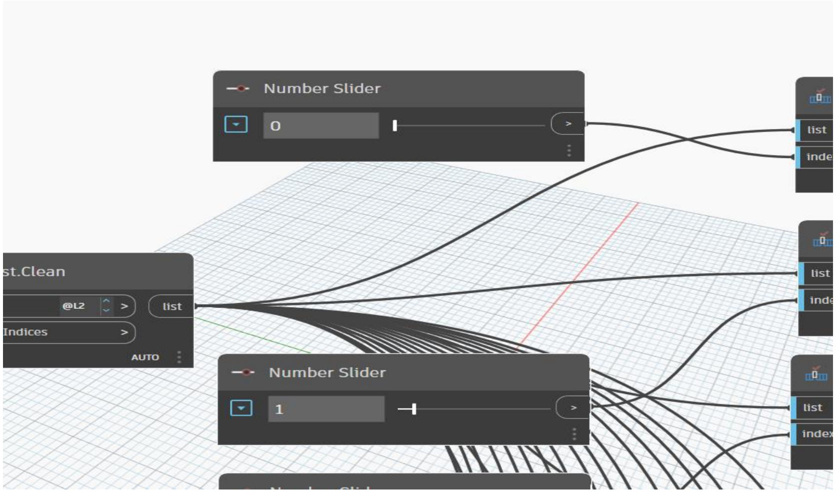
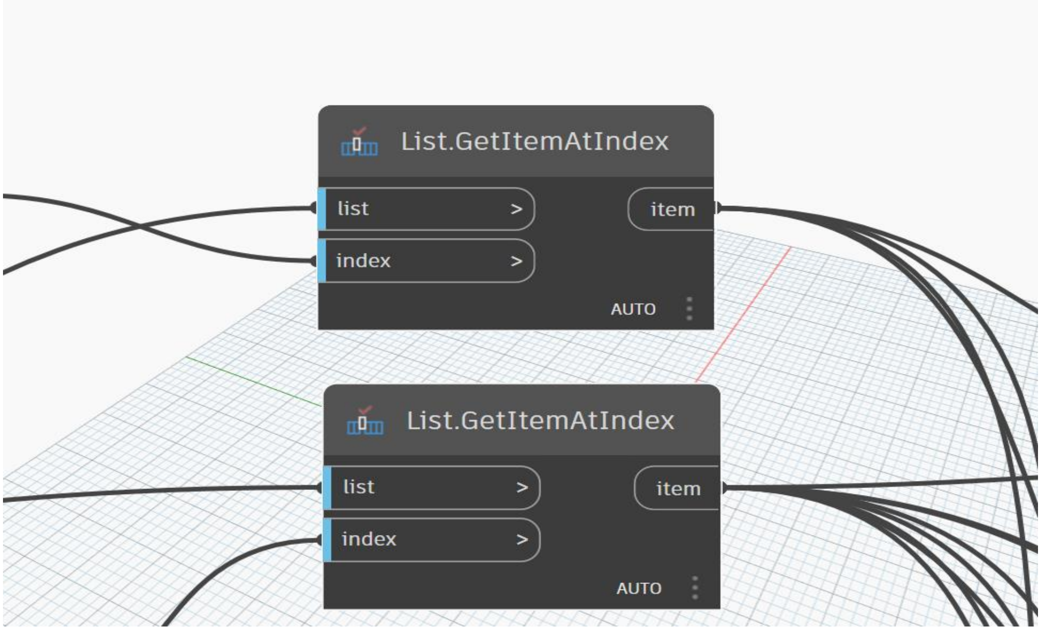


Figure 4.29 – Number Slider

You must create or open a graph in Dynamo in order to use the "List.GetItemAtIndex" node. You can choose the "Get Item At Index" node by going to the "List" tab in the graph editor once you've entered it. By providing its index, this node enables you to obtain a particular item from a list. You can get the desired item as an output by connecting a list and an index value to the node's input ports. When you wish to access and modify particular elements within a list based on their placements, this functionality is helpful. the following List.GetItemAtIndex is shown in figure 4.30



.

Figure 4.30 – List.GetItemAtIndex

In Dynamo, there are a few steps you must take in order to use the FamilyInstance.ByPoint node. Make sure you have the required inputs before you begin: the family type and the location where you wish to deploy the instance. To the node, connect these inputs. The host element and any other appropriate parameters should then be specified. Run the FamilyInstance and the script to finish.At the given location, the ByPoint node will construct an instance of the selected family type. The Revit package must be installed in Dynamo for this node to function properly because it is especially needed for establishing instances of families in Revit. the following Number Slider is shown in figure 4.31

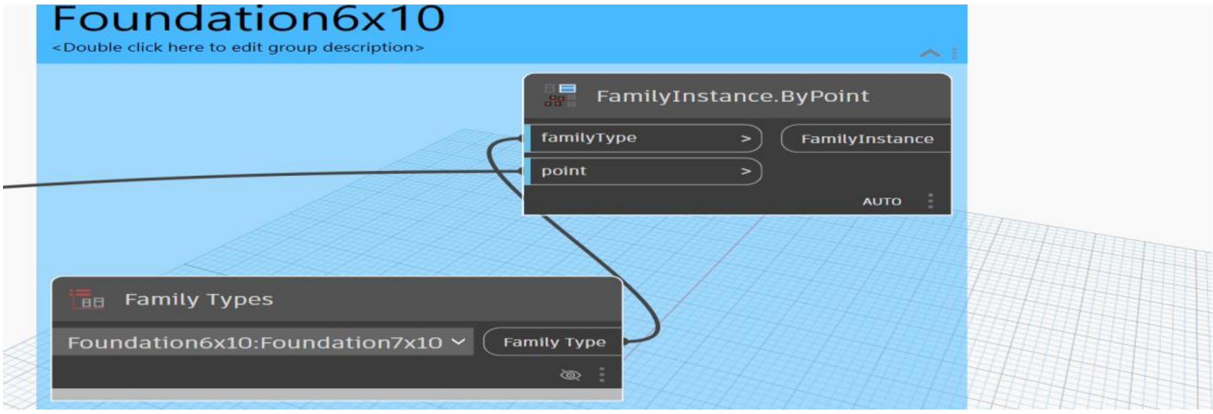


Figure 4.31 – FamilyInstance.ByPoint

Follow these detailed instructions for each node in Dynamo to use it. To List.Enter a list of items and the index of the item you wish to obtain using the GetItemAtIndex function. When using Family.Instance, enter the desired insertion point and the chosen family type to create a new instance. Regarding FamilyInstance.To position an instance at a specific location, use the ByPoint command and enter the family type and insertion point. And now for Line.Enter the start and end points for a line segment using ByStartPointEndPoint. You can efficiently use these nodes in Dynamo to get the outcomes you want by following these procedures. the following Line.ByStartPointEndPoint is shown in figure 4.32.

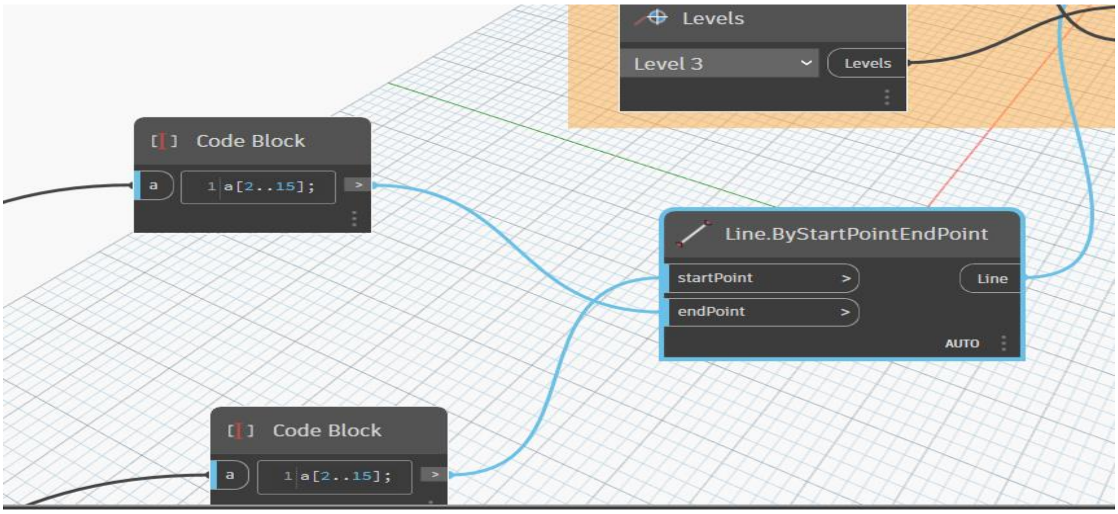


Figure 4.32 – Line.ByStartPointEndPoint

leverage these procedures to leverage Dynamo's StructuralFraming.ColumnByCurve node. Start by drawing a curve with the Line.ByStartEndPoint node or any appropriate tool. Connect the ColumnByCurve node's input to the curve's output next. Connect the relevant family type to the "Type" parameter to specify the desired column type. Run the graph once more, and the node will create a structural framing column along the chosen family type and curve. To achieve the desired column geometry, adjust the inputs as necessary. the following StructuralFraming.ColumnByCurve is shown in figures 4.33.

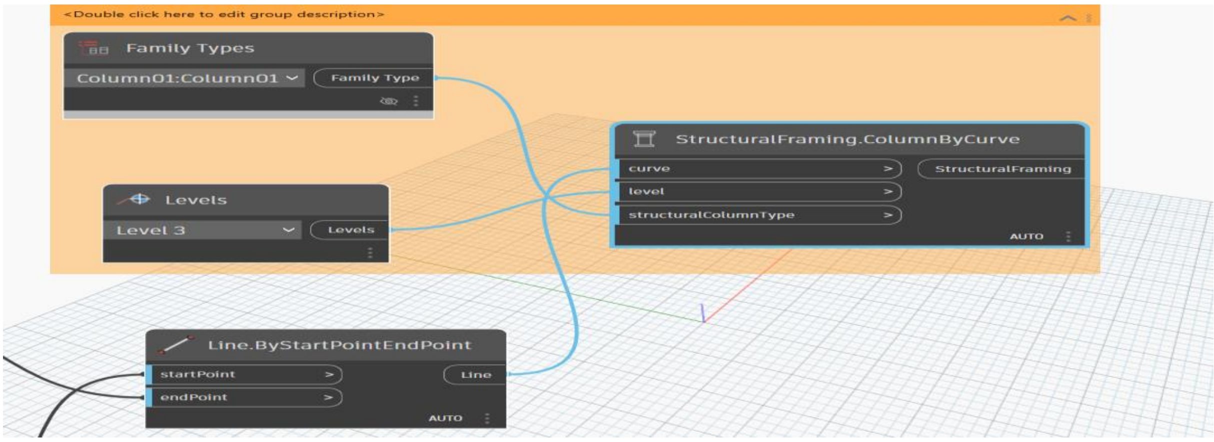
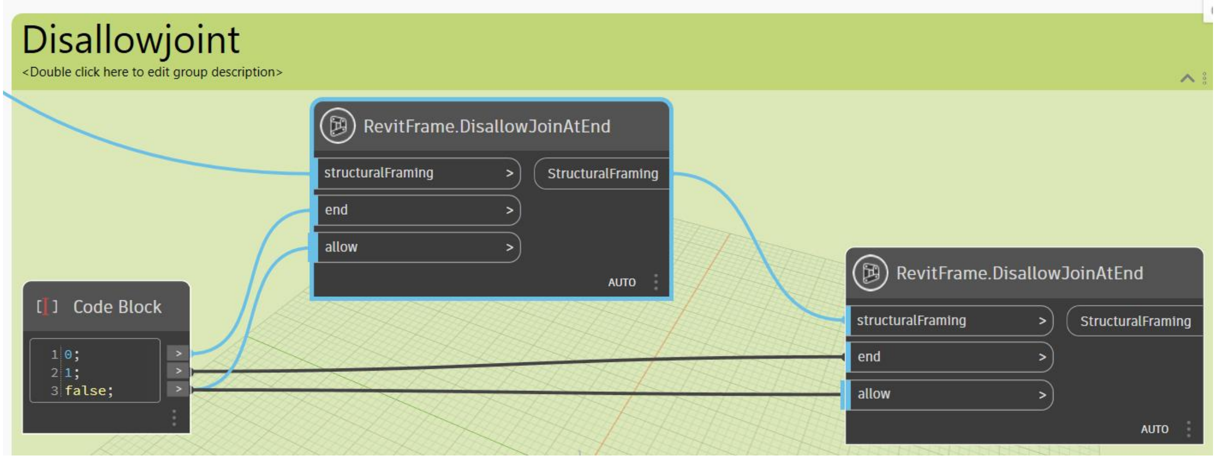


Figure 4.33 – StructuralFraming.ColumnByCurve

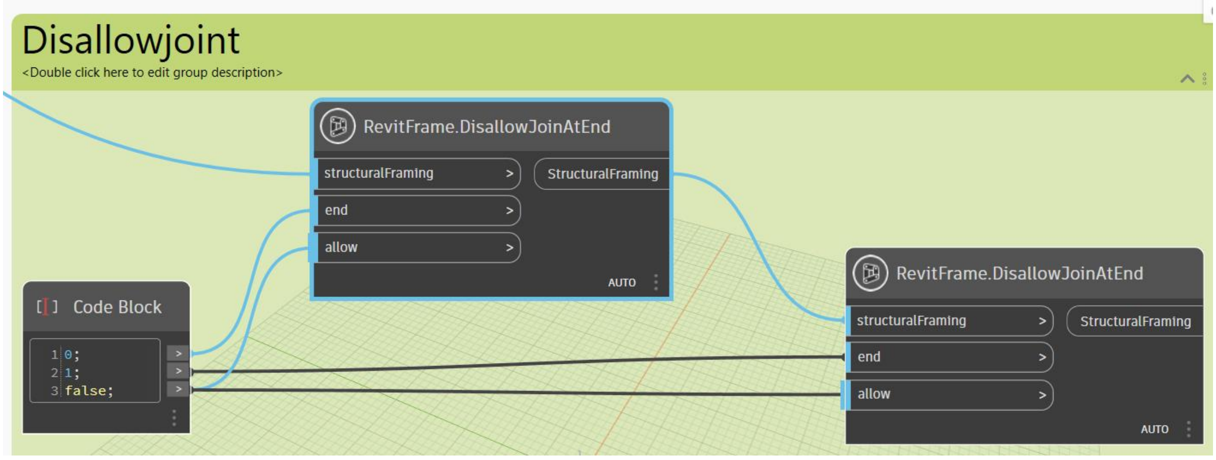
You must supply the required inputs to the "RevitFrame.DisallowJoinAtEnd node" in Dynamo in order to use it.. The output of the structural framing should first be connected.ColumnByCurve node connected to the RevitFrame's input.DenyJoinAtEnd node. the column or beam element for which you want to prohibit joining at the end, and then specify it. You can use this node to stop the end of a structural frame element from connecting to other elements in a Revit model. You may make sure that a specific element doesn't become dependent on any other elements by forbidding the join at the end. the following "RevitFrame.DisallowJoinAtEnd node" is shown in figure 4.34.



```
0;
1;
false;
```

Figure 4.34 – RevitFrame.DisallowJoinAtEnd node

You must enter a list of points that you want to fit a line across in order to use the "Line.ByBestFitThroughPoints" node in Dynamo. This node will automatically determine the line that fits these locations the best. This node will produce a line segment that represents the best-fit line as its output. Line.ByBestFitThroughPoints is a helpful tool for a variety of geometric or data analysis jobs in Dynamo because it is simple to create a line that roughly approximates the trend or pattern of a given set of points. The following Line.ByBestFitThroughPoints is shown in figure 4.35.



```
0;
1;
false;
```

Figure 4.35 – Line.ByBestFitThroughPoints

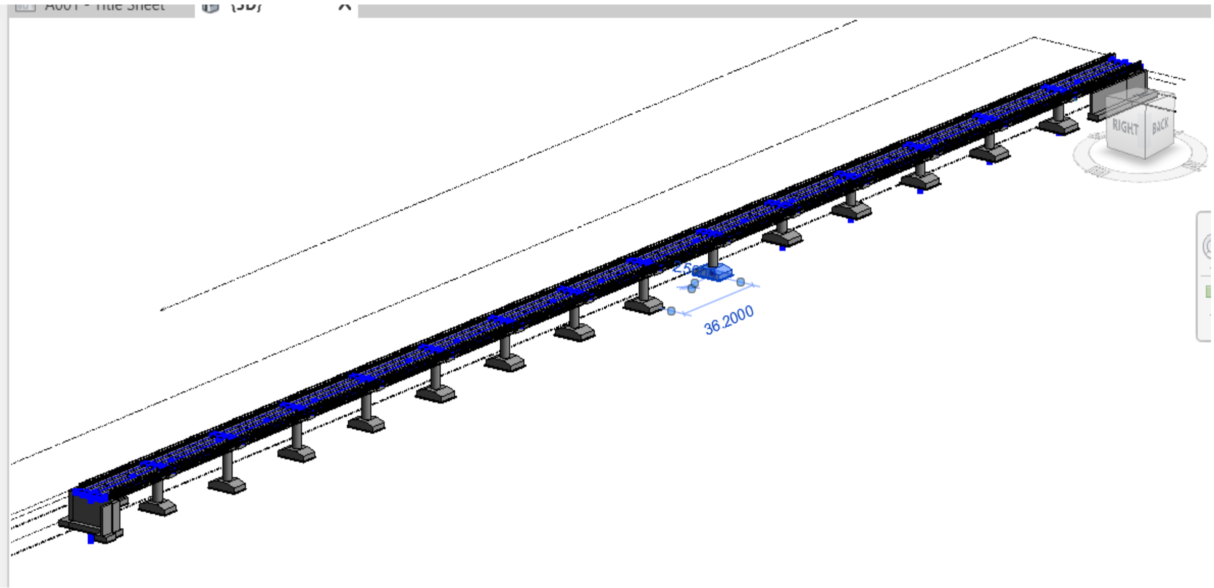


Figure 4.36 – Bridge model

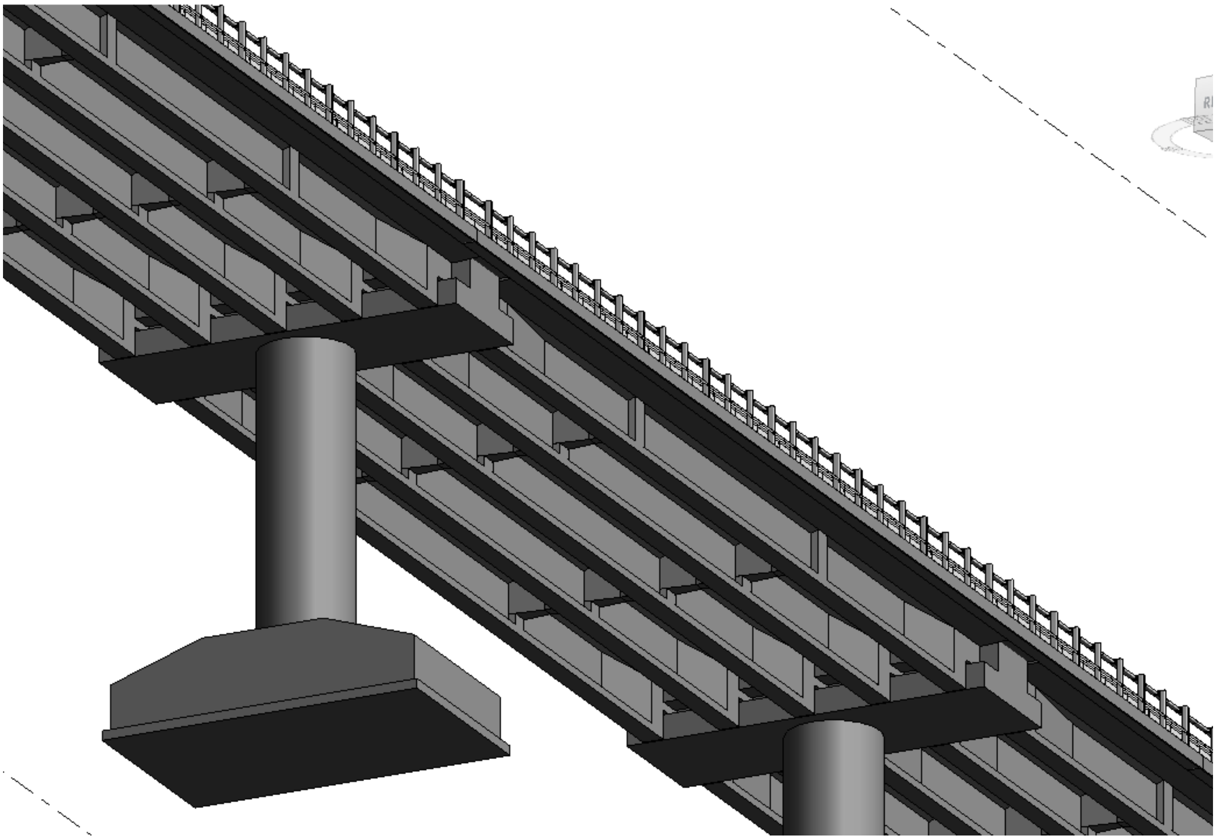


Figure 4.37 – Bridge model

4.2. Visual programming environment

A Visual Programming Environment (VPE) in Revit marks a revolutionary change from the software's conventional design and modeling processes. Through a visual, node-based interface, it offers a cutting-edge method for architects, engineers, and designers to develop intricate parametric designs and workflows. Users no longer need to utilize traditional code or scripting to manipulate design elements in this environment because logic and relationships are represented graphically. Professionals may quickly construct complex design systems while keeping a clear picture of the underlying processes by dragging and linking nodes that correspond to different design parameters and functionalities.

By democratizing the design process, Revit's Visual Programming Environment improves collaboration and efficiency. Because of how much simpler it is to communicate design intent when using a graphical interface, it makes it possible for interdisciplinary teams to collaborate effectively. This system also promotes speedy experimentation and iteration, enabling designers to swiftly change parameters and observe the effects on the design in real-time. As a result, design concepts can be improved and optimized more quickly, enabling better decision-making throughout the course of a project.

The Visual Programming Environment in Revit improves efficiency and collaboration by decentralizing the design process. Due to the interface's graphical form, which makes it simpler to communicate design intent across disciplines, it allows interdisciplinary teams to collaborate easily. This platform also promotes speedy experimentation and iteration, enabling designers to make quick changes to parameters and immediately see how they affect the design. Design ideas can then be improved and optimized more quickly, enabling more informed choices to be made over the course of a project.

Additionally, Revit's Visual Programming Environment encourages a more user-friendly and approachable method of computational design. This node-based method enables designers to develop complex design logic without the need for considerable programming knowledge, in contrast to traditional coding, which frequently requires specialized programming abilities. A wider range of professionals are encouraged to use parametric design thanks to the democratization of design technology and to explore previously unimaginable creative possibilities. Revit's VPE enables architects and engineers to concentrate on the design process itself, fostering innovation and efficiency in their projects, by removing the coding barrier.

4.3. Procedures for applying optimization

An organized strategy is required when utilizing Dynamo in Revit to apply optimization algorithms to a bridge model. Establish the design parameters and optimization objectives first. These could include things like material characteristics, load circumstances, and goals like reducing material utilization. Create a parametric model in Dynamo next, using visual nodes to capture the shape and relationships of the bridge. Include optimization techniques in the model, such as gradient-based or genetic algorithms. These algorithms search out the best solutions by iteratively adjusting design parameters while considering specified constraints.

For designs to be practical, restrictions like material restrictions and safety regulations are essential. Evaluate the effectiveness of each created solution in relation to the goals and restrictions during the optimization process. Structural analysis can assess elements like stability and stress distribution. In an iterative process, settings and algorithms are modified to achieve better outcomes. When the optimization reaches workable solutions, thoroughly assess them. Before choosing the final design, consider elements besides performance, such as constructability and aesthetics. Designers may produce novel, efficient, and safe bridges by combining parametric modeling in Dynamo with optimization. The simple fact that an architect is presented with decisions and options that could have conflicting implications on a building's performance led to the requirement for design optimization (Touloupaki et al,. 2017).

There is a methodical process for integrating optimization approaches into bridge modeling using Dynamo in Revit. First, describe the bridge's design specifications, including its dimensions, construction materials, and other important factors. Establish the optimization goals, such as attaining a certain engineering standard or striking a balance between cost and performance. Create a visual parametric model using Dynamo's node-based framework after that. Create nodes to depict the different design components and their connections. The model should then incorporate optimization techniques that will dynamically modify the design parameters in accordance with the set objectives.

Every iteration of the optimization process generates several bridge designs. These designs are evaluated based on performance standards like structural stability and load-bearing capacity. The solutions must also abide by restrictions like material restrictions and safety standards. The optimization technique improves the design by iteratively adjusting parameters and constraints. This cycle of iterations continues until the algorithm finds a set of best-possible answers that satisfy both engineering requirements and performance objectives. In the end, the optimization inside Dynamo integration for bridge modeling in Revit accelerates the design process, resulting in bridges that demonstrate improved efficiency, durability, and creativity.

4.4. Dynamo script for structural engineering optimization

The process of creating a Dynamo script for structural engineering optimization allows engineers to produce designs that are more reliable and efficient. The parametric modeling and optimization procedures that Dynamo enables in this setting improve the conventional structural engineering technique. Engineers start by determining the important factors and elements that influence the structural design, such as the properties of the materials, the loads, and the geometrical parameters. Engineers build a parametric model that accounts for these variables and their interactions using a node-based visual interface. This model serves as the blank canvas for the application of optimization techniques. A more beneficial approach is presented by BIM methodology, which has the advantages of design coherence and productivity.

The Dynamo script is then updated with optimization algorithms, such as gradient-based techniques and genetic algorithms. These algorithms iteratively investigate multiple design choices by methodically adjusting the design parameters within predetermined limits. The script assesses the structural performance of the design against specified objectives and restrictions during each iteration. To evaluate elements such as stress distribution, deflection, and stability, finite element studies or other simulation approaches may be used. By adjusting the algorithm's parameters and limitations, engineers can cause it to converge towards designs that meet technical requirements and deliver the desired performance results. A BIM model has more information than a typical drawing, which is the primary distinction between the traditional design process and the BIM approach, which is based on 2D drawings (Barbieri et al., 2023).

The Dynamo script is then updated with optimization algorithms, including everything from genetic algorithms to gradient-based techniques. To iteratively investigate different design alternatives, these algorithms methodically modify the design parameters within preset limits. Each iteration of the script compares the design's structural performance against predetermined goals and limitations. To examine elements such as stress distribution, deflection, and stability, this study may involve using finite element analysis or other simulation approaches. The parameters and limitations of the algorithm can be fine-tuned by engineers, allowing the optimization process to converge towards designs that meet engineering requirements while attaining the required performance results. The Architecture, Engineering, and Construction (AEC) sector uses the design and management methodology known as Building Information Modeling (BIM) extensively (Biancardo et al., 2020).

A parametric model that captures important design factors and their interdependencies forms the basis of a Dynamo script. Defining variables like span lengths, member sizes, and load conditions is required for this. The node-based interface of Dynamo makes it easier to set up the parametric model. To create the logic guiding the optimization process, engineers visually connect nodes representing geometric elements, mathematical processes, and conditional statements. Engineers establish optimization goals such as lowering deflection, conserving material usage, and cost optimization. The optimization techniques built within the script are developed with these objectives in mind. The optimization algorithms are chosen and incorporated into the script by engineers. Common algorithms include gradient-based techniques, simulated annealing, and genetic algorithms. In order to accomplish the specified optimization objectives, these methods iteratively alter design variables.

The script is parameterized by engineers by specifying limits and ranges for design variables. To achieve practical and realistic solutions, constraints include material strengths, code compliance, and geometrical limits. By giving the design variables values that fall within the predetermined ranges, the script creates an initial solution. This acts as the point at which the optimization process begins. Beginning with the established optimization objectives, the optimization process is carried out, repeatedly modifying the design factors to enhance the result. Every iteration evaluates how well the design adheres to the limitations. Engineers use structural analysis within the script to assess the effectiveness of each design iteration. Finite element calculations are performed as part of this examination to identify variables including stress distribution, deflection, and stability. An objective function that measures the performance of each design iteration is defined by engineers.

In order to decide when the optimization process should end, engineers establish convergence criteria. Achieving a specific level of performance or the point at which the optimization process no longer generates appreciable improvements are two examples of this condition. Engineers create graphical representations of design variables, performance measures, and the evolution of the design over iterations to visualize the outcomes of the optimization process. Engineers use sensitivity assessments to comprehend how modifications to certain design parameters impact the performance of the entire system. This helps decision-makers understand key variables and how they affect the optimization process. In order to simultaneously optimize for numerous objectives, such as structural effectiveness, cost-effectiveness, and environmental impact, engineers investigate multi-objective optimization for complicated projects.

Platforms for Building Information Modeling (BIM), such as Revit, can be connected with Dynamo scripts. This makes it possible for the detailed design process to segue smoothly from the optimization script. The communication between structural engineers, architects, and other stakeholders is facilitated by dynamo scripts by visualizing the optimization process. The script allows engineers to quickly examine many design scenarios, facilitating effective decision-making and exploring creative design options. The script makes use of data-driven design principles and guides the optimization process and design refinement with real-time analysis results. Engineers use Dynamo's adaptive components to build flexible designs for complicated geometries that can respond to shifting circumstances. Engineers can input design requirements and preferences using a user-friendly interface provided by dynamo scripts, which streamlines the setup process for optimization. Virtual design approaches and technologies can help in the creation of decision bases, however it can be difficult to balance all the trade-offs between various architectural design disciplines (Sandberg et al., 2019).

Engineers assess the robustness of the optimized designs to make sure that they operate consistently under a range of load conditions and uncertainty. In Dynamo, optimization is iterative, resulting in a feedback loop where analysis findings influence design changes that provide better solutions.

Dynamo scripts are used as educational resources because they give students and professionals a practical, visual interface through which to comprehend optimization techniques. Engineers can improve optimization decisions and gain knowledge from previous projects by incorporating prior project data into the script. To determine which factors significantly affect the optimization process and the design outcome, engineers study parameter sensitivity. To assure the accuracy and dependability of the improved designs, engineers validate them using actual physical testing or verified simulation models.

The script can be changed to include resilience factors, optimizing architectures that can resist extreme events while maintaining functioning. To demonstrate the Dynamo script's usefulness in improving diverse structural systems, from bridges to skyscrapers, engineers perform case studies. To create effective designs, the optimization process takes into account the distribution of resources including labor, time, and materials. To better comprehend the trade-offs between various design components, engineers show how limitations affect the design space. The script encourages informed decisions by providing interdisciplinary teams with optimized solutions, enabling collaborative decision-making. Engineers can fine-tune the optimization process in accordance with project requirements because they have control over the optimization parameters.

The optimization process is stabilized by engineers who incorporate error-handling features in the script to deal with unforeseen circumstances. For the benefit of upcoming users and reviewers, engineers document the methodology, presumptions, and reasoning of the optimization script. Engineers illustrate the script's superiority by comparing the optimization results it produces with benchmark designs or conventional techniques. Engineers incorporate manufacturing limitations and considerations into the script to make sure the optimum designs can actually be manufactured. Engineers produce designs that use less energy by taking energy efficiency into account during the optimization process for sustainable projects. Users of the Dynamo script provide feedback, which engineers utilize to iterate and enhance the script. To improve the precision of the optimization process, the script integrates data from numerous sources, including material databases and environmental data. To stay up with new technological developments and emerging optimization methods in the field of structural engineering, engineers are constantly improving and developing the Dynamo script.

An participatory and well-informed design process is promoted by the engineers' real-time input on how changes to design factors affect overall performance.The script assists in risk reduction by spotting potential design flaws and vulnerabilities, allowing developers to take care of these problems quickly. The script's capabilities are used by engineers to investigate novel structure forms that address challenging site circumstances and functional specifications. Engineers can swiftly explore a large design space thanks to the script, which makes it easier to find novel ideas that might not have been considered otherwise. By maximizing structures for occupant comfort, safety, and usefulness, engineers include human-centric design concepts, improving the user experience as a whole.

The script can be altered by engineers to meet project-specific needs, ensuring that the optimization process is in line with the project's particular objectives.The visual character of the script promotes efficient communication and cooperation between numerous disciplines, including mechanical engineering and architecture.In order to manage resource-intensive analyses, engineers use cloud-based compute resources, which enables larger and more intricate optimization studies. The script makes it easier to perform a parametric sensitivity analysis to determine how different design parameters affect the outcomes of optimization.The story fills the gap between design and analysis, advocating a comprehensive strategy in which performance objectives guide design choices.The automatic optimization procedure of the script shortens the design cycle for engineers, resulting in quicker project completion.

Engineers can better understand the variety of options and trade-offs involved in the optimization process by visualizing the design space exploration. The story has instructional value because it gives students practical experience in computational design and optimization, preparing them for problems they would face in the real world. By reducing material waste and building costs, the script's capacity to optimize designs for cost-effectiveness generates major economic benefits. Engineers do post-optimization analysis to confirm that the optimized designs meet performance requirements in a variety of circumstances. The script supports well-balanced and comprehensive solutions by facilitating the resolution of complex trade-offs between competing design objectives.In order to create structures that can adapt to changing conditions and needs, engineers employ a script to foresee future changes and uncertainties.

The script is used by engineers to automate the creation of thorough optimization reports, improving documentation and collaboration. To guarantee that the optimization process complies with established standards and norms, engineers incorporate quality control methods within the script. By introducing engineers to computational techniques and optimization algorithms, the screenplay promotes cross-disciplinary learning. Engineers may more efficiently allocate resources according to the script's improved optimization process, which also enhances project management effectiveness. Engineers use the script to quickly go through many design choices as part of dynamic design exploration. To ensure that the optimal solutions are adaptable to a range of circumstances, engineers investigate numerous design scenarios based on distinct project contexts.

In order to make sure that structures are accessible and inclusive for all users, engineers incorporate accessibility considerations into the optimization process. The script helps extend the lifespan of aged infrastructure by adapting existing structures and optimizing interventions that improve performance. Engineers can better comprehend the optimization process by visualizing the design's evolutionary path through iterations. Engineers can apply sustainable design principles by using the script to optimize structures for minimal energy use and environmental effect. By working on the script simultaneously, engineers practice concurrent engineering, which speeds up decision-making and design iteration. Engineers do risk-benefit studies on various optimization results, balancing any potential benefits against any hazards or uncertainties that may be present.

By giving a visual depiction of design possibilities and their accompanying performance measures, the script improves communication among project stakeholders. By maximizing designs that satisfy both practical and aesthetic objectives, engineers balance structural efficiency with aesthetics. Engineers may identify key locations and guarantee structural integrity by visualizing stress distribution across optimal designs.

In this research, a concrete girder beam is going to be optimized. The inputs are, vertical load, width of beam, length of beam and height of beam. The outputs are cost, maximum bending moment, maximum shear force, maximum displacement.

The below script will optimize the concrete girder beam, which is so important for structural engineering. The following Dynamo script, concrete girder beam and optimization are shown in figures 4.38 to 4.29.

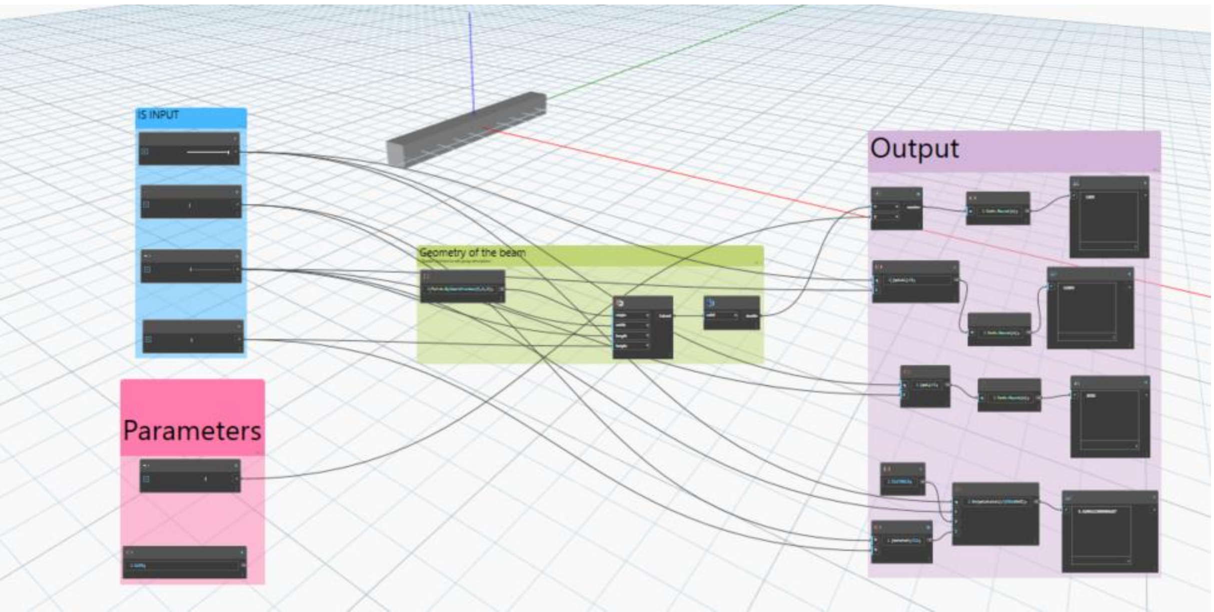


Figure 4.38 – Dynamo script and concrete girder beam

Using text-based scripting, you can create unique expressions in Dynamo using the Code Block node. Drag a Code Block node from the node library onto the canvas to get started. To open the Code Block node's editor, double-click on it. You can create the script in the editor to create points using given coordinates.Use the syntax "Point.ByCoordinates(x, y, z)" in the Code Block to build a point using coordinates. The appropriate numerical values for the point's X, Y, and Z coordinates should be substituted for "x," "y," and "z." For instance, your code block would be "Point.ByCoordinates(5, 10, 3)" to generate a point with the coordinates (5, 12, 6). By adding a semicolon to separate each point creation, you can generate numerous points.

Loops within code blocks can be used to automatically produce points. For instance, you can use a "for" loop to increase the X-coordinate for each point in order to generate a row of points down the X-axis. Variables are supported by code blocks also. For flexibility and readability, define coordinate variables and use them within the "Point.ByCoordinates" function. Code blocks can contain conditional expressions (such as "if" and "else") for more complicated point production based on predetermined standards. You can generate points using mathematical relationships between coordinates since code blocks accept mathematical expressions. Utilizing input from other nodes will allow you to dynamically alter your code. For instance, you can find the coordinates of a point using slider values.

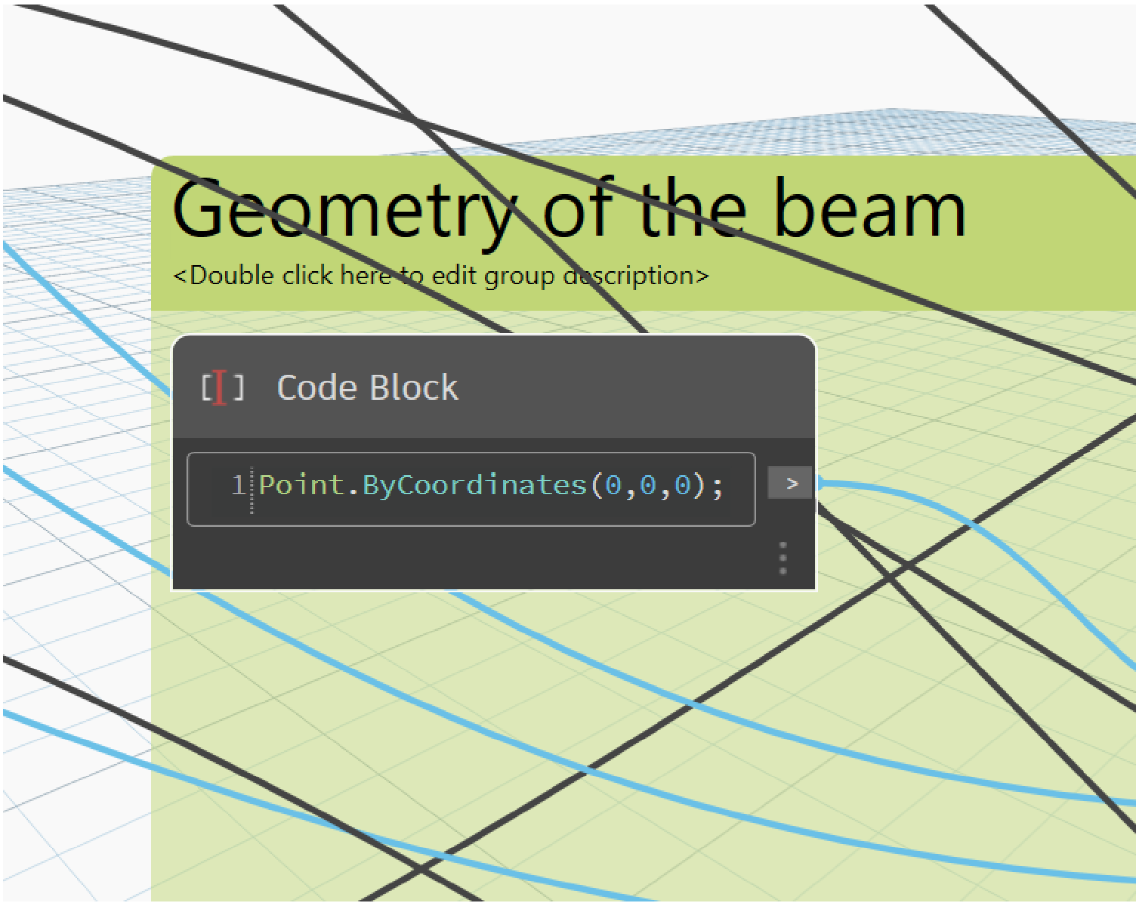
Utilizing input from other nodes will allow you to dynamically alter your code. For instance, you can find the coordinates of a point using slider values.By reading coordinates from the data, you can create points dynamically using external data sources like CSV files. Instead of making individual points, you can declare several sets of coordinates using lists, then use a loop to generate points for each set of coordinates. You can utilize code blocks as a component of more intricate workflows requiring geometric transformations or analysis because they can be nestled inside other nodes. If your code doesn't function as you had intended, Dynamo offers error messages that can help you spot problems and fix them.

Dynamo gives you immediate feedback when you make changes to the code block by changing the preview of the generated points. Once the code block has produced points successfully, you may store it as a custom node and reuse it in other projects. To access extra functions or methods for more complex point generation jobs, Dynamo supports external libraries in code blocks. It is advised to include script comments for difficult code blocks to clarify their purpose and help other users comprehend them. Avoid making logical or grammatical errors that could result in improper point generation. Pay close attention to the error messages that Dynamo displays.

When placing points, keep in mind that the coordinate system you are using—such as global, local, or custom—can have an impact.Dynamic dimensioning is made possible by the use of formulas that incorporate the coordinates of other points to produce new points.Code blocks enable quick design iterations, enabling you to try out various coordinates right away. Share your code blocks with coworkers to promote teamwork and jointly enhance design workflows. Code blocks make repetitive point creation activities automatable, reducing errors and saving time. You can import a list of coordinates from outside sources into Dynamo and use code blocks to turn the coordinates into points. Use vector math inside code blocks to create points depending on angles and separations.

Using loops and mathematical equations, code blocks are able to produce repeating patterns of points. If necessary, you can produce points using code blocks and export the generated coordinates for use in other programs. Create procedural or randomized layouts by employing code blocks to introduce unpredictability into point generation. Dynamo updates the point placements in real time to provide you visual feedback as you make changes to the code. To locate new nodes and functions that can improve your point generating scripts, look into Dynamo packages. To efficiently use code blocks, Dynamo offers a wealth of online tools, tutorials, and documentation. If your code doesn't yield the desired outcomes, consider disassembling it to find the potential source of the problem.

Code blocks facilitate the iterative design process, allowing you to fine-tune point generation until you get the desired result. The points created while utilizing Dynamo with Revit can be applied in a variety of ways to your building model. Experiment with more complex point creation strategies as you gain confidence with code blocks, and deepen your knowledge of Dynamo scripting. The following code block is shown in figure 4.39.



```
Point.ByCoordinates(0,0,0);
```

Figure 4.39 – Code block

For Cuboid.ByLengths in Dynamo Make a list of the coordinates you intend to utilize to build cuboids. These coordinates ought to be in the Point format.WithCoordinates(x, y, z). Choose the sizes of the sides for each cuboid you wish to construct. For each cuboid, these lengths can be fixed or variable. Move the Cuboid around.Upon the canvas, ByLengths nodes. Connect the lengths list to the Cuboid's Lengths input.Node for ByLengths. Connect the coordinates list to the Cuboid's Center input.Node for ByLengths. To see the cuboids in the Dynamo workspace, enable the preview option. By pressing the "Run" button, the script will be run and the cuboids will be generated using the supplied lengths and positions.

When the script executes correctly, you'll see the cuboids with the required lengths produced at the supplied positions. To generate other configurations, you can alter the input coordinates or cuboids' lengths and run the script again. A combination of List can be used if you have a list of coordinates and lengths.The Cuboid and the map.Use the ByLengths node to simultaneously create several cuboids. If your coordinates and lengths lists are nested, you can use the List.Map node twice: once to iterate over the outer list (coordinates), and once inside that list (lengths). By applying extra nodes and properties, you can further alter the cuboids' characteristics such as material, color, or rotation angles.

You can utilize the Geometry to scale or transform the cuboids.Geometry and scale.Transform the nodes as well as the Cuboid.Node for ByLengths. You can use nodes like File if your coordinates originate from a CSV file or other external source.Data and reading.To extract the data, use fromCSV. Think about utilizing data architectures effectively. If your list of coordinates is lengthy, it may be advisable to use vectorized operations to enhance the efficiency of your script. Make sure the input data is precise and well-structured to prevent script execution issues. Verify the accuracy of the lengths and coordinates before attempting to create cuboids. Unexpected outcomes can emerge from negative lengths or bad coordinate formats.

Verify that the nodes and functionality indicated are supported by the version of Dynamo you're running. Use breakpoints and debugging tools in Dynamo to troubleshoot and find problems if the script doesn't behave as planned. Adjust the lengths and coordinates iteratively to experiment with various arrangements and optimize the final design. To learn more sophisticated methods and innovative ways to use the Cuboid, look through online guides, discussion boards, and documentation for Dynamo.Node for ByLengths. As you gain experience using Dynamo, you might consider creating unique Python scripts inside of Dynamo to increase the utility and adaptability of your cuboid generating procedure. To prevent mistakes or unexpected results, be sure that your input data is accurate and reliable.

To make it simple to experiment with various versions, think about incorporating length and coordinate parameters.If you want to analyze the data quickly and generate a lot of cuboids, think about employing list management approaches. You can use Dynamo Player or a custom node to automate the cuboid generating process if your input data changes frequently. To ensure precise placement of cuboids, be aware of the coordinate system you're using, whether it's project-, family-, or global. Share your Dynamo script if you're working in a group, and work together to hone and enhance the cuboid generating procedure. To aid in reproducing or modifying the procedure, keep a record of your scripts, including input information, parameters, and outcomes. The following code block is shown in figure 4.30.

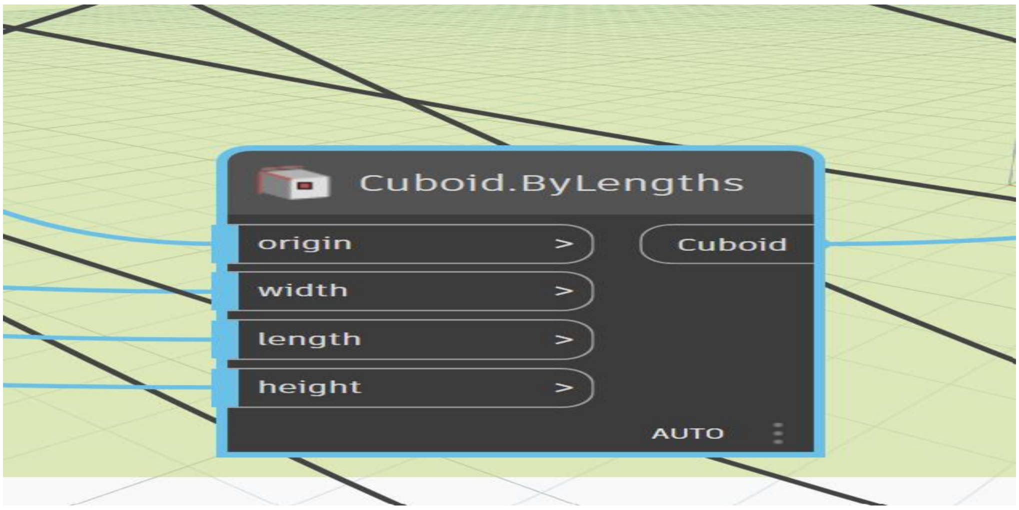


Figure 4.30 – Cuboid.ByLengths

You can define points inside a given volume using coordinates by using the "Solid.Volume" node in Dynamo. A Solid.Volume node should be added to the canvas. affix the Solid.Volume node connected to the solid's input.Node for volume. The 3D bounding box or region that you want to create points for is specified by this input. Connect a list of X, Y, and Z coordinates to the Solid's "Point" input.Node for volume. The volume selected will have points created using these coordinates. Connect the Solid.Volume node's output to the Point output. Based on the input coordinates and the designated volume, this will generate points.

The points can be joined to a geometry.Point node for the Dynamo workspace to display the generated points. Change the Solid's settings.To adjust the size and location of the volume in which points will be generated, use the volume node. To alter the volume's dimensions, change the minimum and maximum corner points. Make a list of the XYZ coordinates you intend to employ for placing points inside the given volume. A point's location is represented by each set of XYZ coordinates. Coordinates can be manually entered by typing them into a list, or they can be taken from other sections of your Dynamo script, external files, or data sources.

You can utilize iterative techniques, like loops or recursion, to input coordinates and generate points dynamically if you have a lot of data to process. The volume dimensions or coordinates can be parametrized to increase the adaptability and flexibility of your script. This enables you to generate points based on several coordinate systems or within various volumes. By adding noise to the coordinates or by picking random points from your coordinate list, you can add randomization to the point production process. You can limit the creation of points in your Dynamo script to certain areas of the volume or in accordance with predetermined criteria by using conditional statements and logic.

To accomplish certain effects, you can alter the coordinates of the input before creating points using coordinate transformations like translation, rotation, or scaling. Visualize the created points in the Dynamo workspace and utilize additional nodes to troubleshoot or analyze the data to make sure your script performs as planned. Depending on the number of points and the complexity of your script, you might want to think about enhancing performance by reducing the amount of computation or data processing that is required. Your script will be easier to comprehend for you and others who may work with it in the future if you comment on it and label it. Implement error handling processes to deal with potential problems like incorrect coordinates or input data. The following code block is shown in figure 4.41.

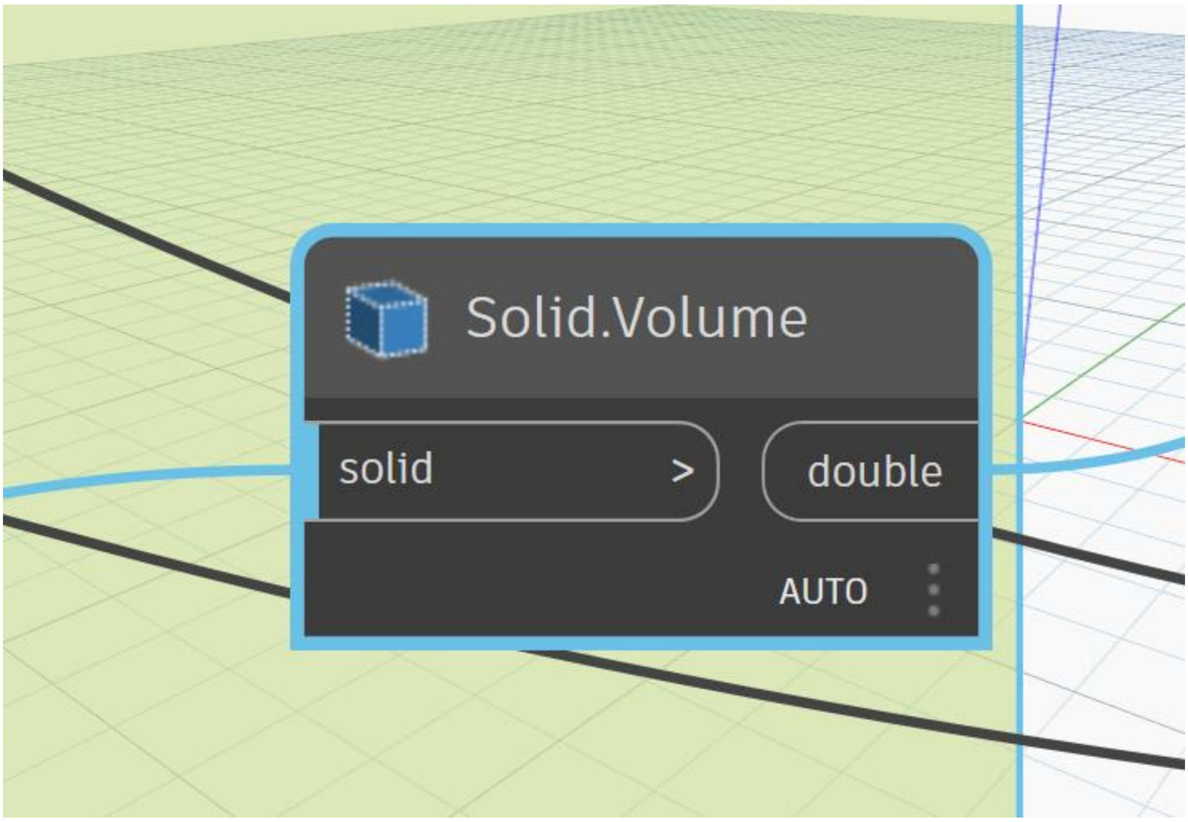


Figure 4.41 – Cuboid.ByLengths

A innovative method of design called generative design has been increasingly popular in recent years. Autodesk Dynamo, a potent visual programming environment that enables architects, engineers, and designers to explore novel solutions through computational design and automation, is one of the well-known platforms for generative design. Users of Dynamo can build parametric models that can generate a large number of design options based on particular factors, producing more effective, optimal, and innovative results.

Dynamo's generative design methodology relies on input parameters like restrictions, objectives, and design criteria to function. These variables direct the algorithm as it generates numerous design alternatives. Following that, the algorithm iteratively assesses and develops these possibilities, modifying and adjusting them in accordance with the preset principles. Through this iterative process, solutions may be found that might otherwise go unnoticed when employing conventional design techniques.

Users of Dynamo may create intricate generative workflows without having to have a deep understanding of coding because to its visual programming interface. Users can join nodes to form relationships and processes by manipulating them graphically. With the help of this strategy, architects and engineers can explore design options while working in a collaborative atmosphere. The platform's adaptability fosters experimentation and pushes the limits of what is possible in terms of design.

Different sectors use Dynamo's generative design technology. By producing optimum floor plans, exterior designs, and interior layouts, it can help architects design structures that are both structurally solid and visually beautiful. By iteratively assessing load-bearing capacities and material utilization, technology can help engineers design lightweight and efficient buildings. Additionally, generative design helps product design by producing original configurations and shapes that satisfy functional specifications.

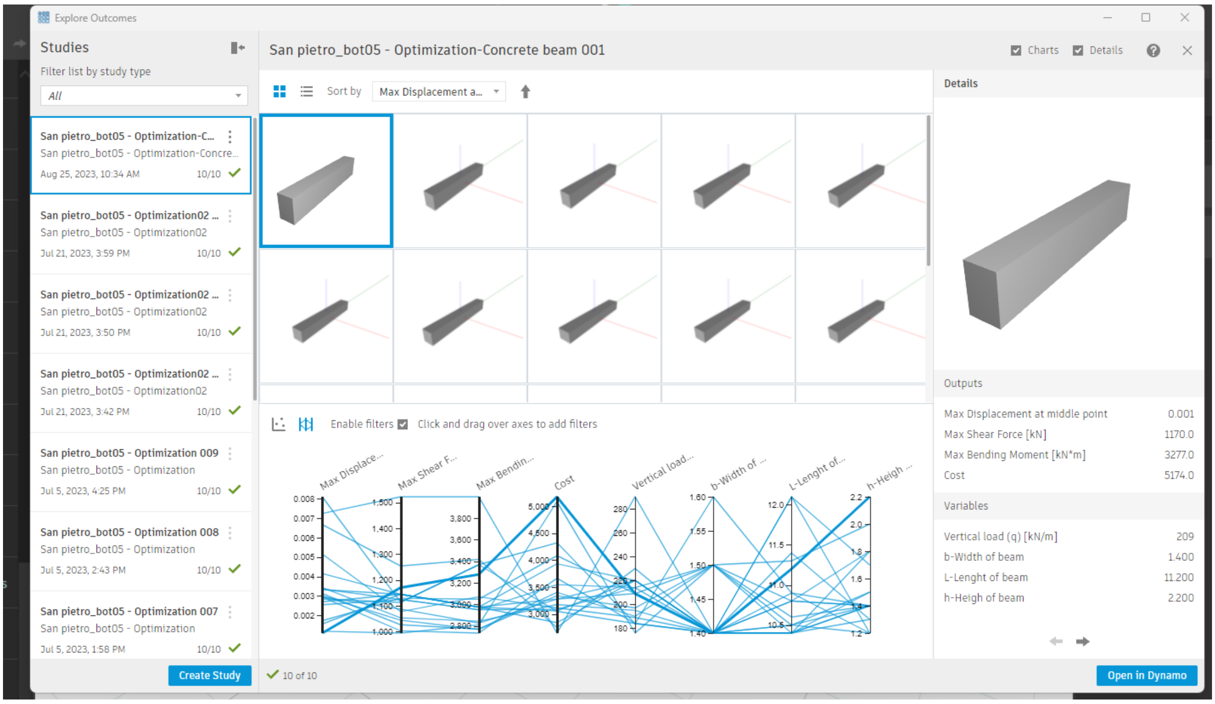
The capacity of generative design in Dynamo to consider several design restrictions at once is a big advantage. The algorithm may consider all of these parameters, including material limitations, financial restrictions, and environmental considerations, ensuring that the resulting solutions are practical in the actual world. This all-encompassing approach aids in the creation of unique and useful designs. Additionally, generative design improves environmental initiatives. Dynamo can find solutions that use the fewest amount of resources, energy, and waste by examining various design iterations. This is in line with the increased focus on environmentally friendly design techniques and helps create a built environment that is more sustainable.

Generative design, meanwhile, is not without its difficulties. The algorithm generates many choices, thus deciding which design is best might be a challenging procedure. In assessing the generated alternatives, taking into consideration any non-quantifiable aspects, human intuition and skill are essential.

Despite its potential, generative design in Dynamo is not a substitute for the ingenuity and knowledge of people; rather, it is a tool to enhance and extend the abilities of designers and engineers. The human touch is still crucial for fine-tuning and choosing among the solutions that were created because it considers aspects like cultural context, user experience, and emotional resonance that automated algorithms could miss.The wide collection of community-contributed plugins and Dynamo's open design allow for further capability expansion. These plugins can offer particular functions that let users expand and modify the generative design process in accordance with their own requirements. Professionals from a variety of fields can use Dynamo to meet their particular projects' needs thanks to its adaptability.

In Dynamo, collaboration becomes essential to generative design workflows. The seamless flow of ideas and data is made possible by the collaborative work of architects, engineers, urban planners, and other stakeholders on a single platform. This quickens the design process and encourages interdisciplinary collaboration, leading to more comprehensive and well-rounded design solutions.The potential of generative design to redefine conventional aesthetics is an interesting feature. The iterative nature of the process frequently produces surprising and unusual design solutions that put traditional ideas about form and structure to the test. This has the potential to usher in a new era of architectural and design innovation, pushing the bounds of what is possible in the built world. Iterative, expensive computational processes can be carried out by a visual programming-based generative design system. (Korus et al., 2021)

In conclusion, Dynamo's generative design offers the potential to completely alter how we develop and design. Designers can explore a large design space and find solutions that are both creative and useful by making use of computation, algorithms, and collaborative tools. The platform's versatility is highlighted by its capacity to take into account various limits and its adaptation to various sectors. While issues like decision-making and the importance of human intuition still exist, there are several chances to push the boundaries of creativity and problem-solving. As generative design develops, it is expected to help create a future in which technology and human intellect coexist together to create our reality. The following generative design is shown in figures 4.42.



Outputs

Max Displacement at middle point 0.001
Max Shear Force [kN] 1170.0
Max Bending Moment [kN*m] 3277.0
Cost 5174.0

Variables

Vertical load (q) [kN/m] 209
b-Width of beam 1.400
L-Lenght of beam 11.200
h-Heigh of beam 2.200

Figure 4.42 – Generative design

This page is intentionally left blank

5. DISCUSSION

5.1. Results interpretation in relation to research questions

The research questions act as a considering clipless that direct the study of the data that is gathered and validate the results of the optimization process. The results show that visual programming tools like Dynamo may be used to generate intricate parametric models, answering the first study question about the efficiency of visual programming in bridge optimization. The analysis shows that the visual interface does, in fact, make it simpler to create complex design reasoning. The outcomes also show how visual programming enables engineers and designers to explore creative design solutions that might otherwise be difficult to implement using conventional techniques, regardless of their level of coding knowledge. This result supports the idea that visual programming is a game-changing strategy for bridge optimization.

The second study question examines how optimization methods affect how effectively bridges are designed. It becomes clear from the results interpretation that adding optimization methods greatly improves the design process. The analysis demonstrates how the used algorithms systematically change design parameters, encouraging the investigation of design options that adhere to predetermined goals and restrictions. This is consistent with the basic hypotheses that optimization algorithms are essential for improving bridge designs because they let engineers investigate a variety of options and converge on the best ones.

The third study topic examines how well the optimization results adhere to safety requirements and structural engineering norms. By interpreting the findings in this light, it is clear how essential limits are to guaranteeing the viability and security of optimized solutions. The findings show that while optimization algorithms excel at developing effective designs, the inclusion of restrictions ensures that these designs comply with important structural and safety requirements. This realization supports the idea that optimization calls for strict adherence to industry standards in order to ensure that the pursuit of efficiency does not jeopardize structural integrity.

5.2. Studying restrictions

First and foremost, it's critical to comprehend the visual programming's limits as an optimization modeling tool. Visual programming may have some limitations in terms of the complexity and flexibility of the models that may be produced, even though it can provide a user-friendly interface and an easier coding process. Therefore, it is essential to carefully examine these constraints and specify the research's scope. The second step in examining constraints is to pinpoint the precise optimization issues that visual programming can solve. Numerous fields, including resource allocation, scheduling, and supply chain management, can be included in optimization models. The most effective and practical ways to apply visual programming techniques to the selected problem domain must be identified, along with the types of optimization problems that lend themselves to this approach.

It is crucial to comprehend how restrictions affect structural integrity. To guarantee that the optimization process does not jeopardize the bridge's structural integrity, restrictions are examined in relation to variables including load-bearing capacity, stress distribution, and stability. Furthermore, taking into account the accessibility and availability of visual programming tools is necessary for this dissertation's examination of constraints. It's critical to evaluate the variety of visual programming frameworks and platforms that are currently available and that can be applied to optimization modeling. To choose the best platform for the research and make informed judgments, it is crucial to comprehend the strengths and weaknesses of different instruments. It's also critical to take into account the software's usability, learning curve, and interoperability with other tools and libraries that are needed.

Last but not least, researching restraints for this dissertation should also look at any prospective hardware or software constraints. Visual programming tools could need specific hardware requirements or software dependencies, which might limit the produced models' usability and scalability. It is crucial to examine these specifications and make sure the research can be carried out with the infrastructure and resources at hand.

It is crucial to comprehend how restrictions affect structural integrity. To guarantee that the optimization process does not jeopardize the bridge's structural integrity, restrictions are examined in relation to variables including load-bearing capacity, stress distribution, and stability. International and national building codes must be followed when designing bridges. The optimized models are compliant and can be smoothly integrated into actual construction projects without running afoul of any laws or regulations thanks to research on the limitations associated with these codes. In order to create the best possible bridge designs, limitations on the types and qualities of the materials employed are essential. Examining how material constraints impact the results of the design might reveal useful information about the viability and sustainability of the optimized structures.

It is crucial to comprehend how restrictions affect structural integrity. To guarantee that the optimization process does not jeopardize the bridge's structural integrity, restrictions are examined in relation to variables including load-bearing capacity, stress distribution, and stability.

International and national building codes must be followed when designing bridges. The optimized models are compliant and can be smoothly integrated into actual construction projects without running afoul of any laws or regulations thanks to research on the limitations associated with these codes. In order to create the best possible bridge designs, limitations on the types and qualities of the materials employed are essential. Examining how material constraints impact the results of the design might reveal useful information about the viability and sustainability of the optimized structures.

Bridge designs should be in line with the project's required functionality and intended aesthetics. It is made sure that the optimization method does not ignore the project's design intent and user needs by looking into these limits. Collaboration between diverse disciplines is necessary for bridges. Studying constraints from several angles, such as civil engineering, architecture, and urban planning, promotes an all-encompassing optimization strategy that takes into account various project factors.

5.3. Comparative evaluation of currently available optimization tools and techniques

The results of this assessment will shed light on the advantages, disadvantages, and applicability of certain tools and methods for the selected research field. Researchers can choose the best tools and strategies for optimization by carefully taking into account the following considerations.

First and foremost, it's critical to comprehend visual programming's constraints in the context of optimization modeling (erau.edu, n.d.). The complexity of the models that can be developed may be impacted by limitations on the usefulness or scalability of visual programming tools. The degree to which each tool can solve these restrictions and offer a suitable environment for optimization modeling will be revealed by conducting a comparative analysis.

Second, researchers should concentrate on pinpointing certain optimization issues that can be successfully solved with visual programming tools. In some problem fields, such as linear programming, integer programming, or network optimization, certain tools may perform better than others. Researchers can choose the best solutions for their research by comparing the capabilities of each instrument and determining whether it is compatible with the recognized optimization problems.

Additionally, it is important to take into account the accessibility and availability of visual programming tools. The research process could be hampered by the limited accessibility or requirement for specialist technology or software of some technologies. Researchers can choose tools that are widely accessible and work with their study setting by doing a comparative analysis of the accessibility and availability of various resources.

Finally, it's critical to look into any software or hardware restrictions related to the chosen optimization tools. Some tools could be computationally intensive or incompatible with certain operating systems or programming languages. Researchers can anticipate future obstacles and make wise choices about their study methods by assessing these limits.

6. CONCLUSIONS

6.1. Research objectives

The goal of this research is to transform the discipline of bridge engineering. Engineers and designers can use the power of computational optimization to produce creative and effective bridge designs by creating a specific visual programming framework. This method takes advantage of the interaction between algorithmic investigation and human expertise, ultimately resulting in the construction of bridges that are structurally solid, commercially feasible, and aesthetically pleasing.

The validation approach, which will put the suggested visual programming framework through rigorous testing and improvement through a number of real-world case studies, is at the heart of this project. These case studies will cover a wide range of bridge types, demonstrating the framework's adaptability to different design contexts. The study aims to quantify the benefits of adopting this novel methodology by contrasting the results of the visual programming-based optimization approach with conventional methodologies. Additionally, this research will offer recommendations and best practices for successfully incorporating visual programming and optimization into the bridge design procedure, guaranteeing that specialists in the field may implement new developments without difficulty into their workflows.

The dissertation's ultimate goal is to establish a paradigm shift in how bridges are designed and built. The project aims to open up new opportunities in structural design that can result in bridges that are not only more functional but also more creatively designed, by embracing the power of visual programming and optimization techniques. This study aims to accelerate beneficial changes in the field of bridge engineering by a thorough analysis of the research objectives, creating a future where technology and design innovation intersect naturally.

6.2. Suggestions for future research

As you begin your research, you might investigate a number of intriguing directions for additional study that can help advance this discipline. Examine and contrast various optimization techniques as they relate to bridge modeling. Investigate strategies for getting the best bridge designs, including, Examine and contrast various optimization techniques as they relate to bridge modeling. Investigate strategies for getting the best bridge designs. Increase the scope of your research to address several competing goals at once, such as structural effectiveness, cost-effectiveness, and environmental sustainability. Create approaches that can strike a balance between these goals in order to produce well-rounded bridge designs. Learn more about the seamless integration of visual programming platforms like Dynamo with structural analysis software. Investigate techniques to automate the data transfer between these tools so that performance assessments can be made in real time while the optimization process is being carried out.. Increase the scope of your research to address several competing goals at once, such as structural effectiveness, cost-effectiveness, and environmental sustainability. Create approaches that can strike a balance between these goals in order to produce well-rounded bridge designs. Learn more about the seamless integration of visual programming platforms like Dynamo with structural analysis software. Investigate techniques to automate the data transfer between these tools so that performance assessments can be made in real time while the optimization process is being carried out.

Examine the impact of design parameters on how well optimized bridge models perform. This could entail analysing the effects of changes in input parameters on results and investigating techniques to improve robustness in the face of uncertainties. Examine how the designer makes decisions and uses their intuition in a visual programming environment. Examine the effects of interface design, node interactions, and feedback systems on the designer's capacity to successfully steer the optimization process. Look at the possibility of incorporating machine learning methods to improve the optimization process. Create models that include lessons from earlier design iterations to produce initial designs for optimization that are more well-informed. Detailed case studies of bridge projects should be conducted. Analyse the effects of optimization through visual programming on the structural performance, construction efficiency, and maintenance of project outcomes.

Consider the advantages of interdisciplinary cooperation in modeling optimization bridges. Investigate how visual programming may aid in communication and group decision-making by working with architects, civil engineers, urban planners, and other stakeholders. Look into how generative design methods can be incorporated into the visual programming environment. Examine how the system can facilitate the optimization process by automatically generating and evaluating a wide range of design possibilities. For bridge optimization, evaluate the usability and user experience components of visual programming environments. Determine the difficulties that designers might encounter, make suggestions for interface enhancements, and create rules for efficient interaction. Investigate the moral ramifications of automated optimization techniques. Examine questions of human agency, accountability, and the negative effects of automated decision-making in bridge design.

Examine how visual programming can be used to include optimization into engineering teaching. Create instructional resources such as tutorials, workshops, and training modules to impart this novel methodology to the upcoming generation of engineers. Keep an open mind while changing your research's focus in response to new trends, technology, and the demands of the engineering and design community. Your contribution has the potential to make a substantial impact on critical infrastructure optimization and the development of contemporary design methodologies.

REFERENCES

Literature

Korus, K., Salamak, M. and Jasiński, M. (2021). ‘Optimization of geometric parameters of arch bridges using visual programming FEM components and genetic algorithm. Engineering Structures’, [online] 241, p.112465. doi:https://doi.org/10.1016/j.engstruct.2021.112465.

Salamak, M., Jasinski, M., Plaszczyk, T. and Zarski, M. (2019). ‘Analytical Modelling in Dynamo. Transactions of the VŠB – Technical University of Ostrava, Civil Engineering Series, 18(2). doi:https://doi.org/10.31490/tces-2018-0014.

Shim, C.-S., Dang, N.-S., Lon, S. and Jeon, C.-H. (2019). ‘Development of a bridge maintenance system for prestressed concrete bridges using 3D digital twin model’. Structure and Infrastructure Engineering, 15(10), pp.1319–1332. doi:https://doi.org/10.1080/15732479.2019.1620789.

Chong, A.Chen, J.Tapley, M. (2021) ‘Creating Parametric Design Workflows for Rapid Conceptual Design and Optioneering’, 10.2749/christchurch.2021.0912.

Touloupaki, E, Theodosiou, T (2017) ‘Performance Simulation Integrated in Parametric 3D

Modeling as a Method for Early Stage Design Optimization’, Energies 2017, 10, 637; doi:10.3390/en10050637

Barbieri, G., Giani, M., Enrico De Panicis, Biagi, A., ‘Dario Della Femina and Gianluca Di Bella (2023). BIM and Tunnelling – a Norwegian application: the Sotra Link Project’. Proceedings of the 2nd World Congress on Civil, Structural, and Environmental Engineering. doi:https://doi.org/10.11159/icgre23.123.

Biancardo, S.A., Capano, A., de Oliveira, S.G. and Tibaut, A. (2020). ‘Integration of BIM and Procedural Modeling Tools for Road Design. Infrastructures’, [online] 5(4), p.37. doi:https://doi.org/10.3390/infrastructures5040037.

Hartung, R, Schönbach, R, Liepe, D, Klemt-Albert, K. (2020). ‘Automatized Parametric Modeling to Enhance a data-based Maintenance Process for Infrastructure Buildings’, 10.22260/ISARC2020/0038.

Jinjin, S (2021) ‘Application of BIM modeling technology in Bridge Engineering’, IOP Conf. Ser.: Earth Environ. Sci. 781 032028

Sandberg, M., Mukkavaara, J., Shadram, F. and Olofsson, T. (2019). ‘Multidisciplinary Optimization of Life-Cycle Energy and Cost Using a BIM-Based Master Model’. Sustainability, 11(1), p.286. doi:https://doi.org/10.3390/su11010286.

Del Grosso, A, Basso, P ,Ruffini, L (2017) ‘Infrastructure management integrating SHM and BIM procedures’, Fourth Conference on Smart Monitoring, Assessment and Rehabilitation of Civil structures

Sources

www.comune.pontesanpietro.bg.it. (n.d.). Comune Ponte San Pietro: Dati geografici. [online] Available at: https://www.comune.pontesanpietro.bg.it//territorio/dati-geografici [Accessed 7 Sep. 2023].

erau.edu. (n.d.). Research Projects | Embry-Riddle Aeronautical University. [online] Available at: https://erau.edu/research/projects?&l=DB&c=Graduate&co=DCOA&page=1 [Accessed 7 Sep. 2023].

LIST OF ACRONYMS AND ABBREVIATIONS

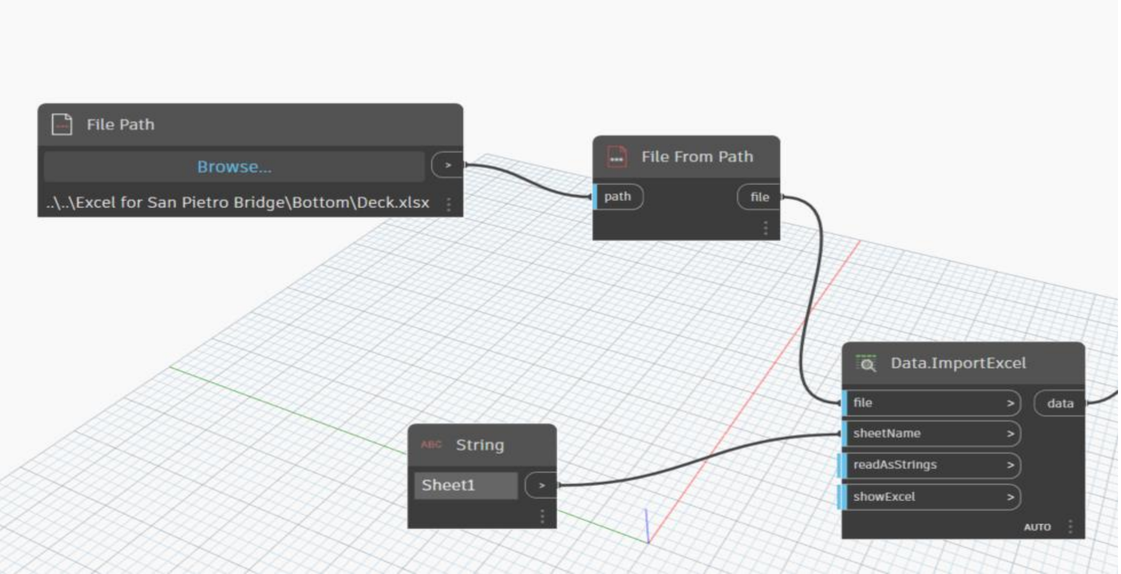
BIM	Building Information Modelling
AEC	Architecture, Engineering and Construction
CAD	Computer aided Design

This page is intentionally left blank

APPENDICES

APPENDIX 1: DYNAMO SCRIPTS

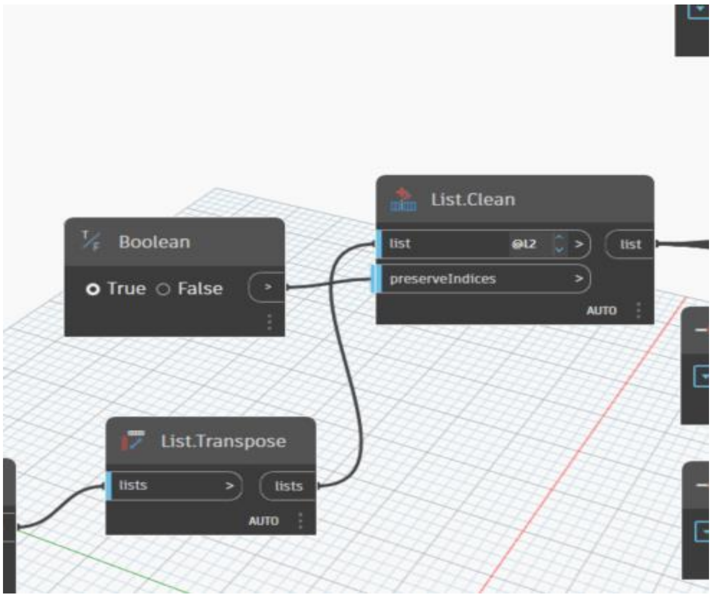
The following Dynamo script models the entire bridge.



File Path: ..\..\Excel for San Pietro Bridge\Bottom\Deck.xlsx

String: Sheet1

Figure 5.43 – Dynamo script



Boolean: ● True ○ False

Figure 5.44 – Dynamo script

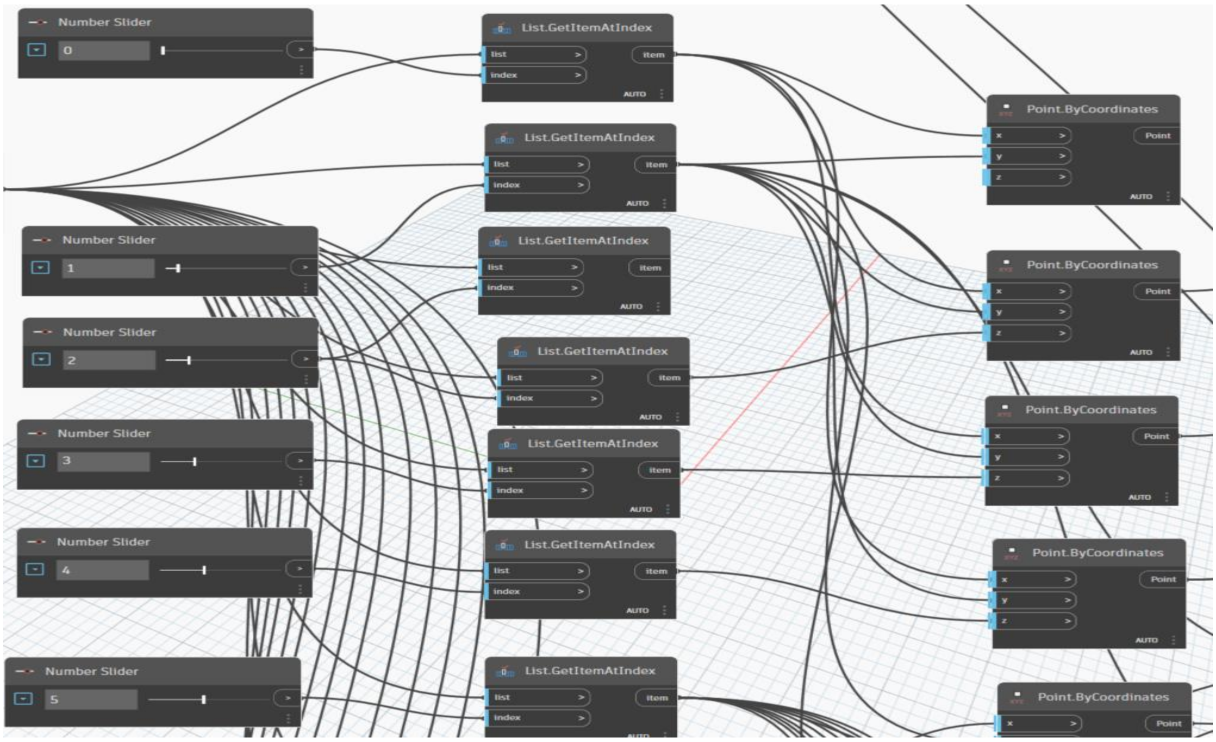


Figure 5.45 – Dynamo script

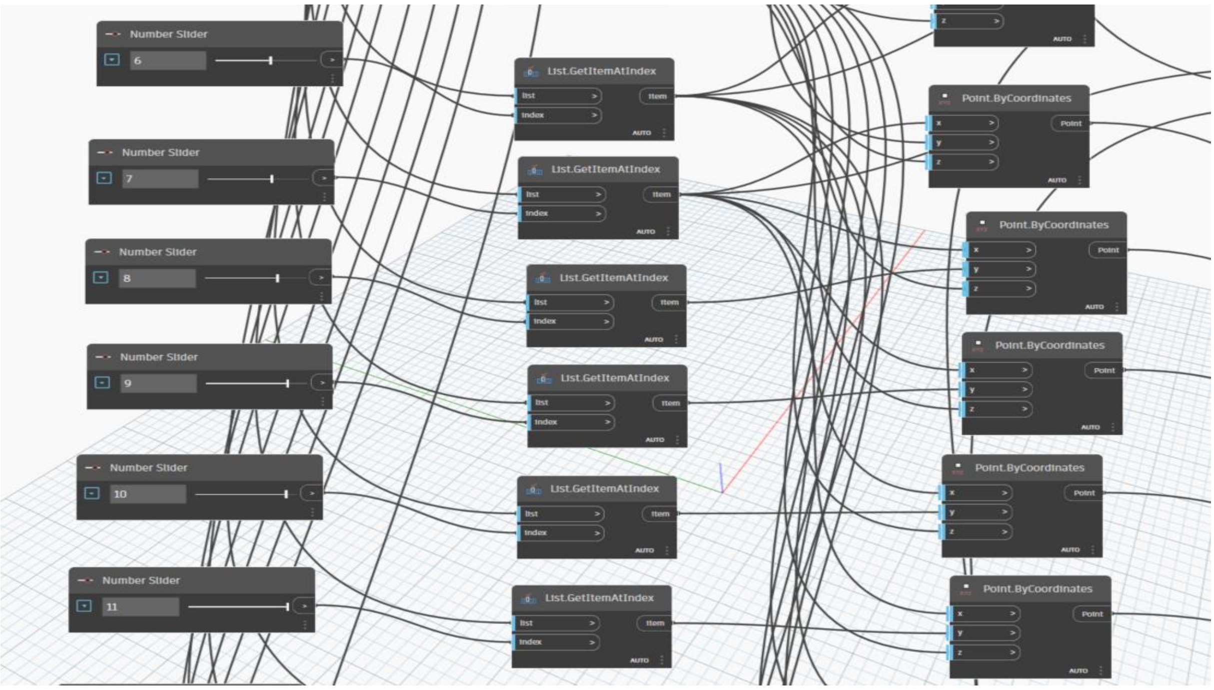


Figure 5.46 – Dynamo script

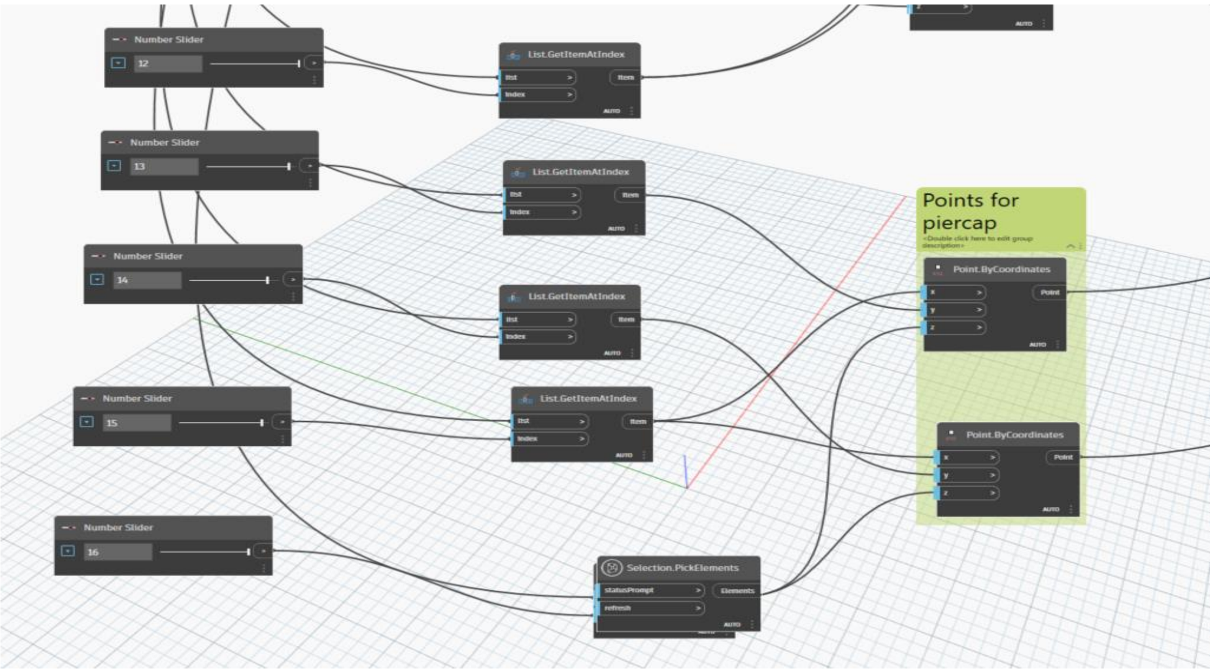


Figure 5.47 – Dynamo script

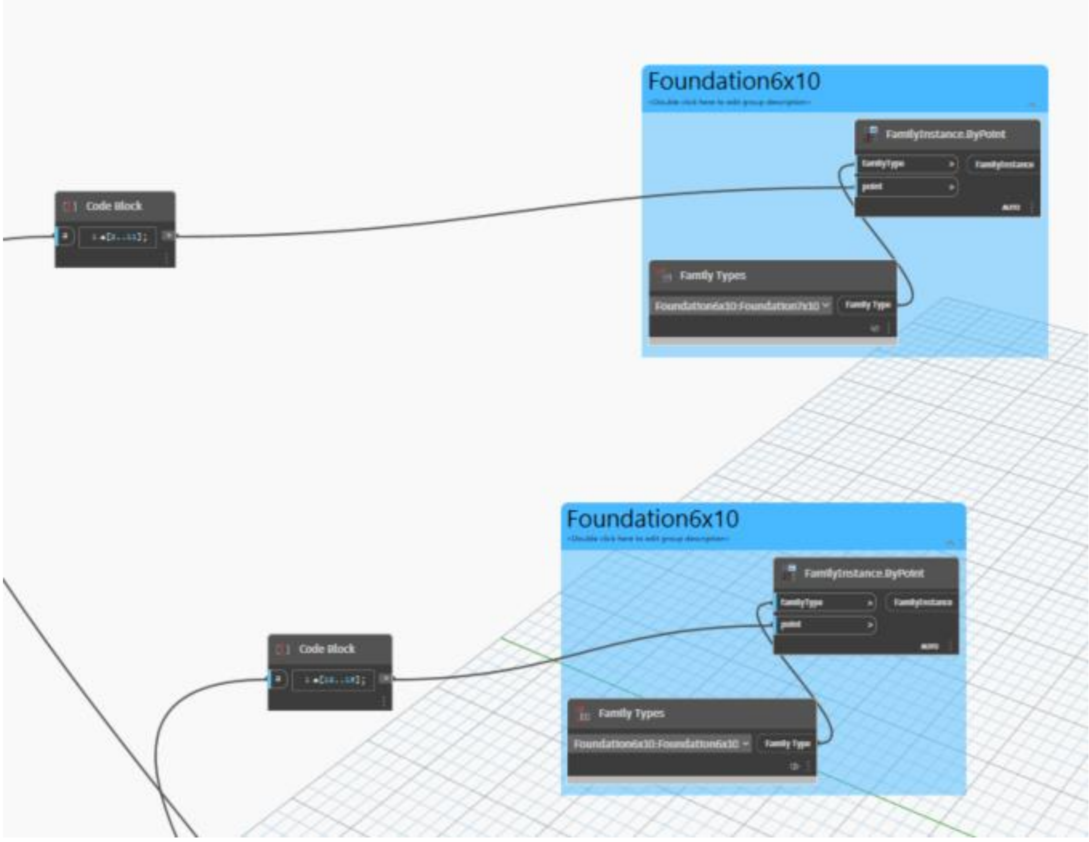


Figure 5.48 – Dynamo script

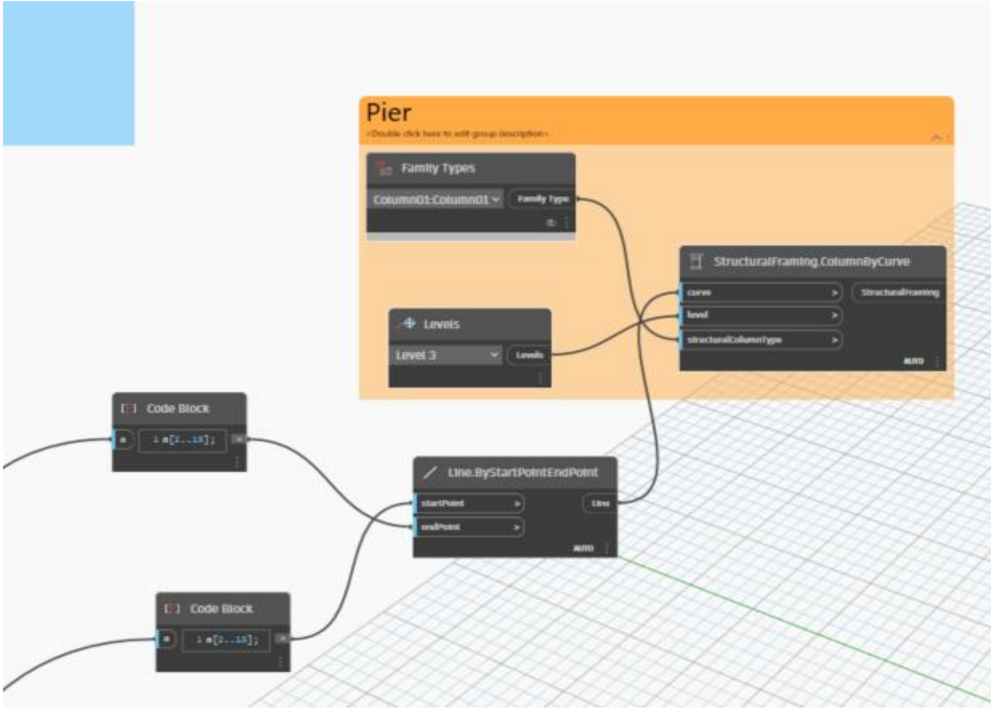


Figure 5.49 – Dynamo script

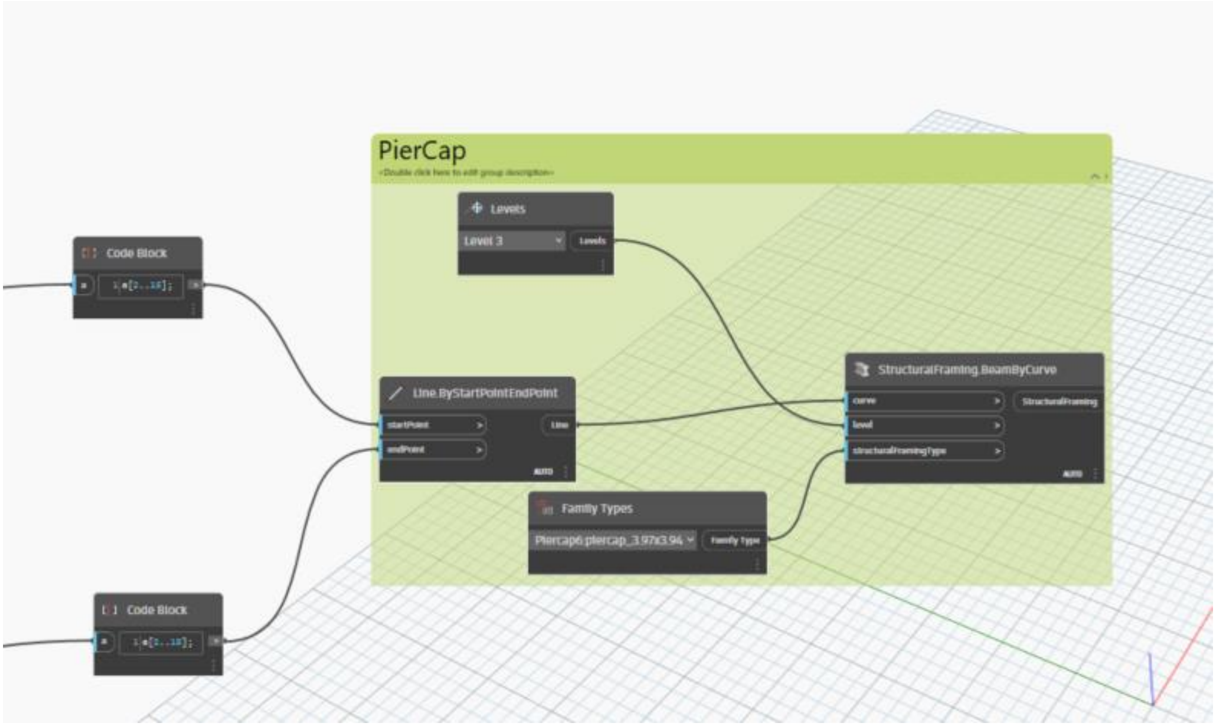
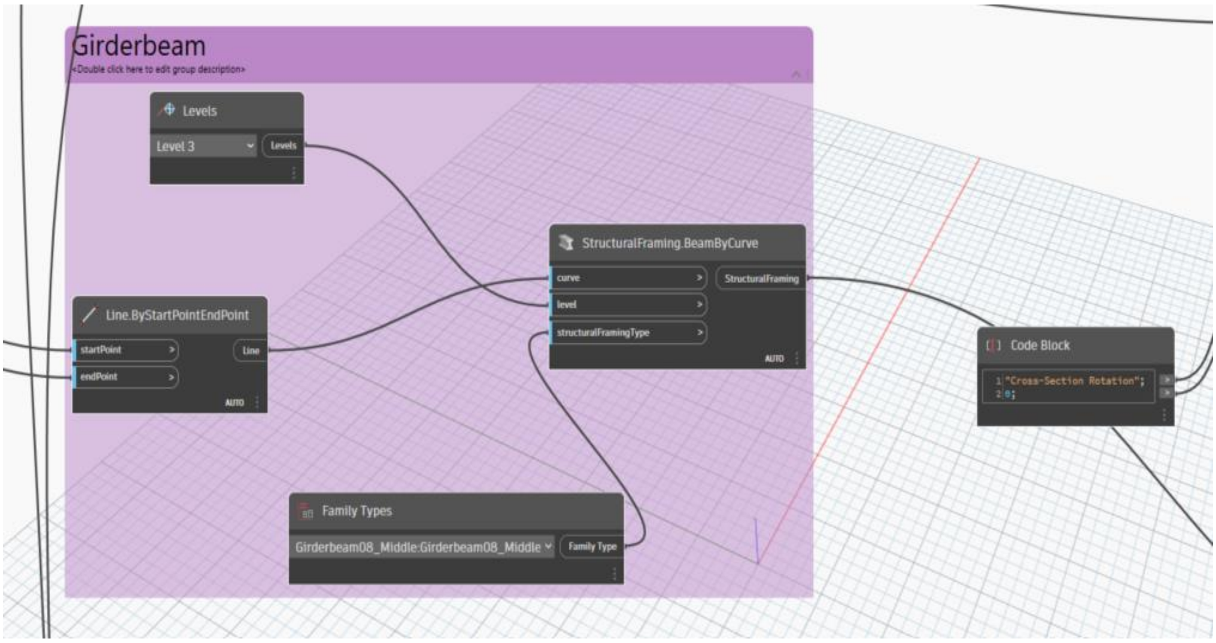
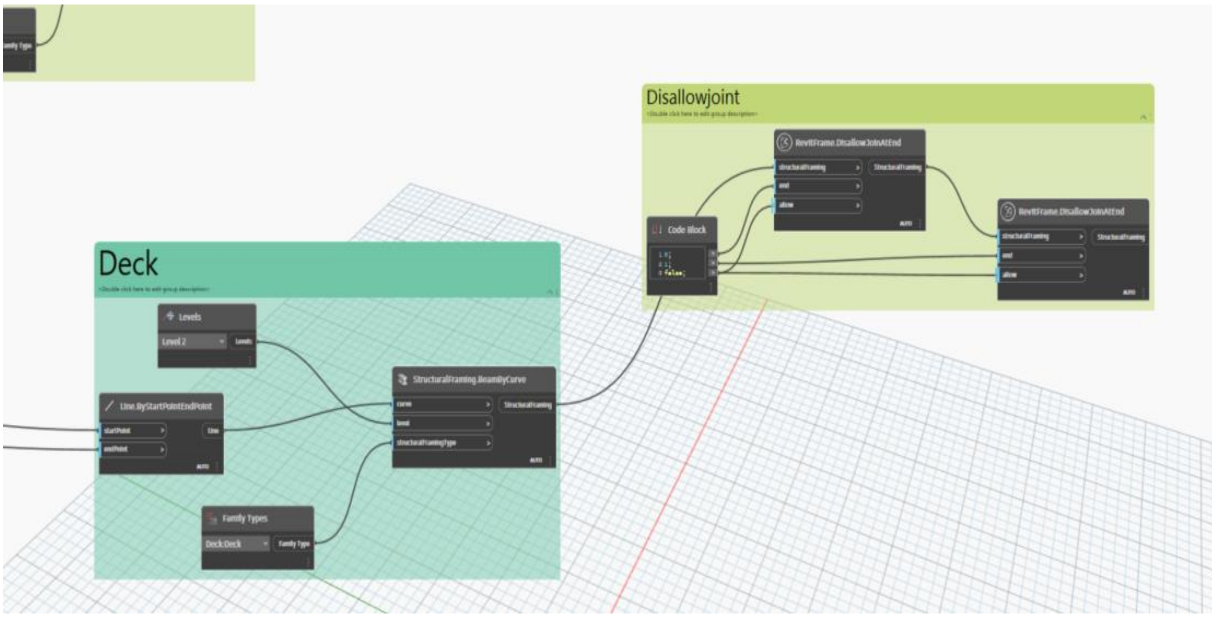


Figure 5.50 – Dynamo script



```
"Cross-Section Rotation";
0;
```

Figure 5.51 – Dynamo script



```
0;
1;
false;
```

Figure 5.52 – Dynamo script

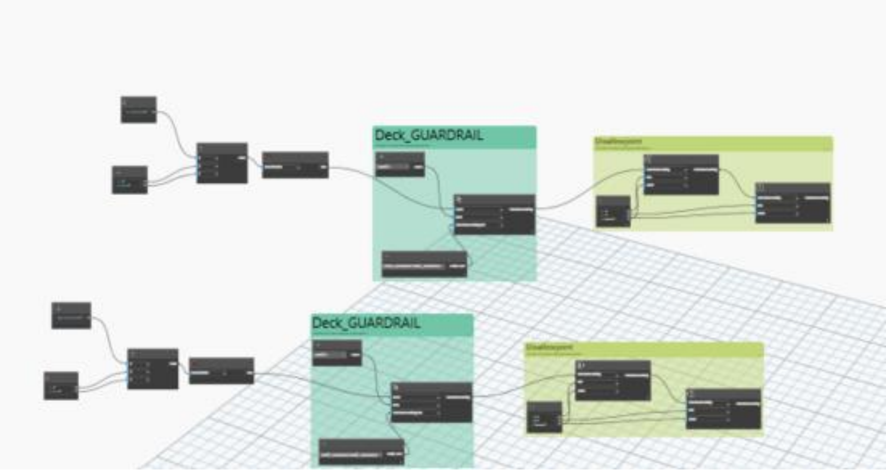


Figure 5.53 – Dynamo script

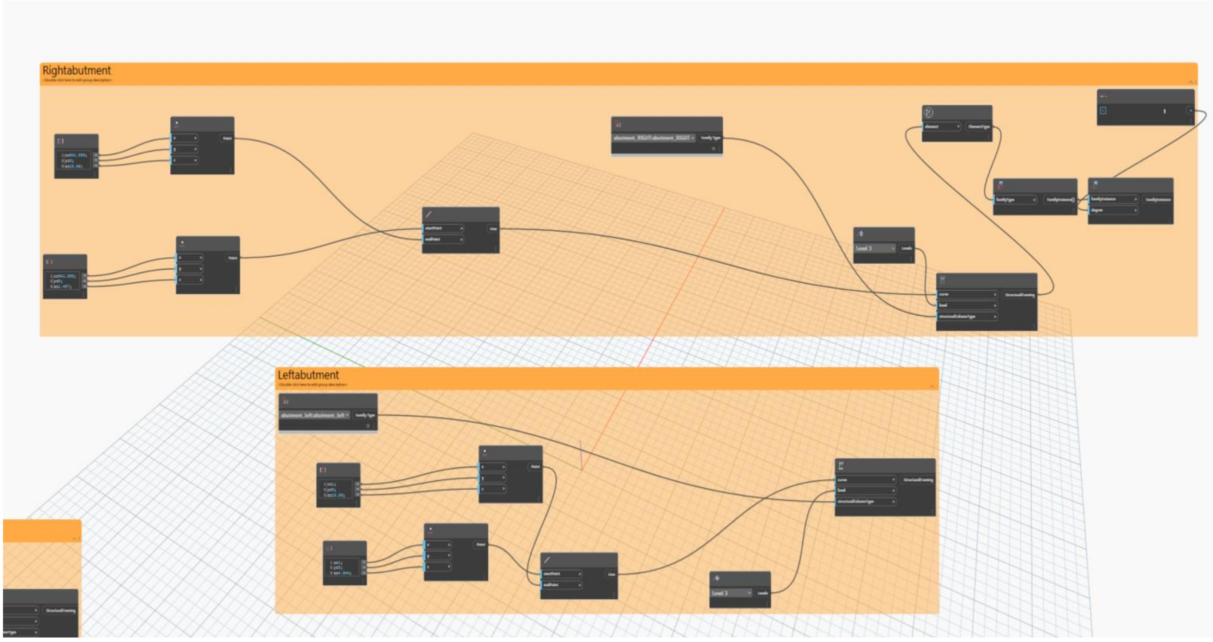
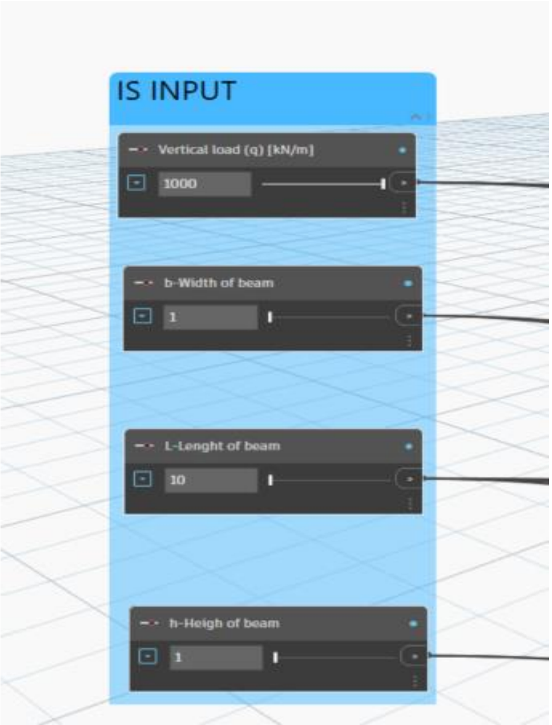


Figure 5.54 – Dynamo script

The following Dynamo script is for structural analysis.



IS INPUT

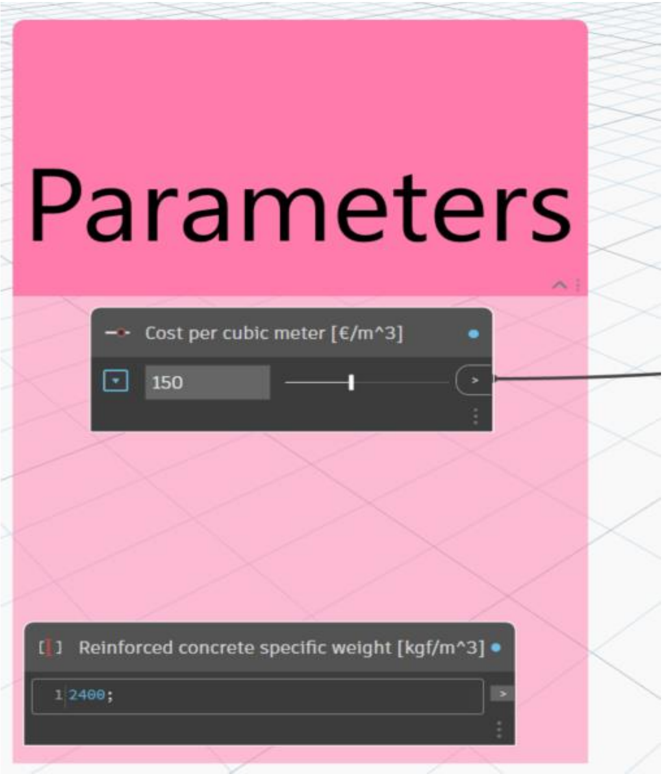
Vertical load (q) [kN/m]: 1000

b-Width of beam: 1

L-Lenght of beam: 10

h-Heigh of beam: 1

Figure 5.55 – Dynamo script



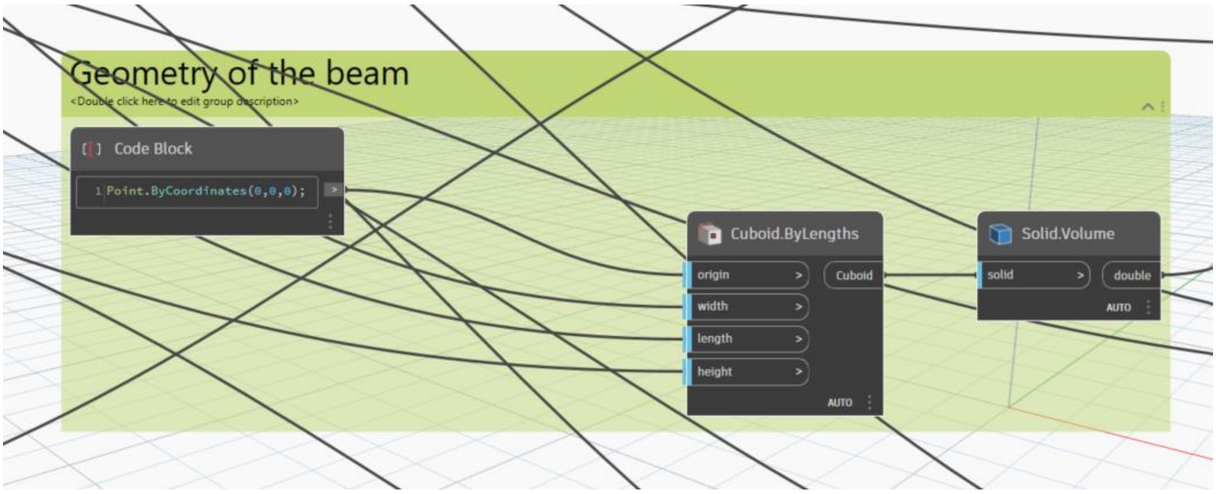
Parameters

Cost per cubic meter [€/m^3]: 150

Reinforced concrete specific weight [kgf/m^3]

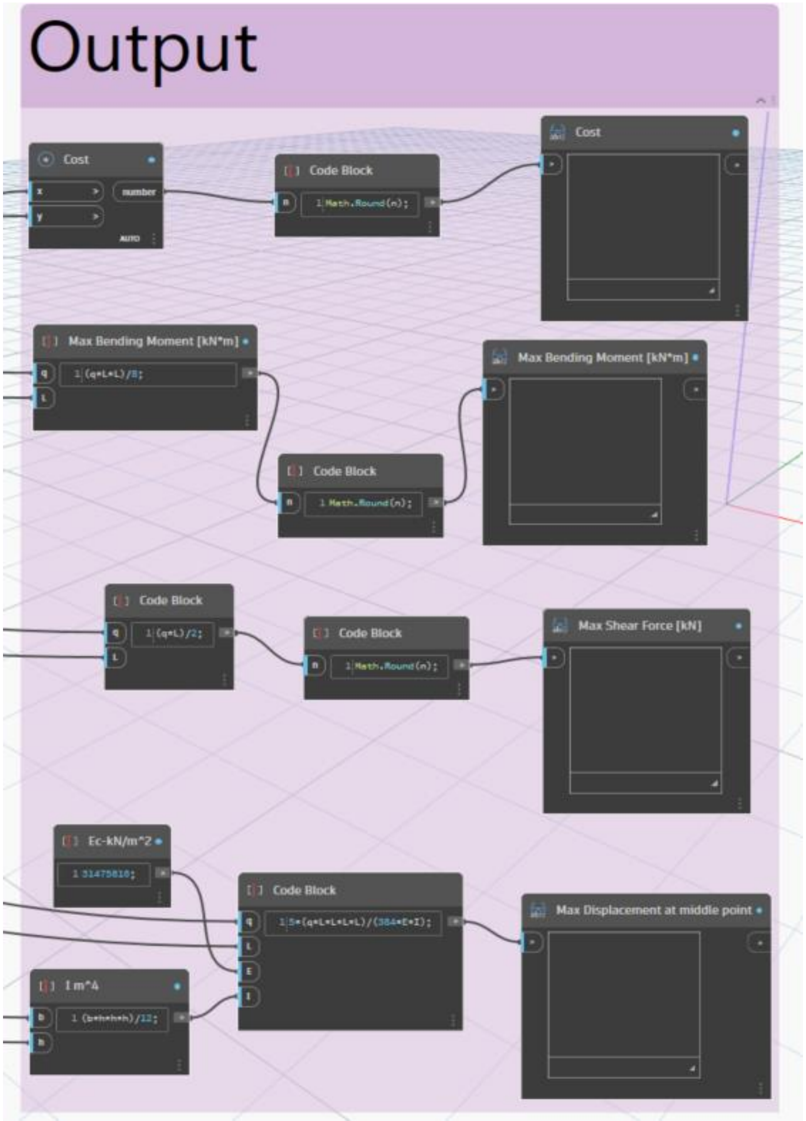
```
2400;
```

Figure 5.56 – Dynamo script



```
Point.ByCoordinates(0,0,0);
```

Figure 5.57 – Dynamo script



```
Math.Round(n);
(q*L*L)/8;
Math.Round(n);
(q*L)/2;
Math.Round(n);
31475810;
5*(q*L*L*L*L)/(384*E*I);
(b*h*h*h)/12;
```

Figure 5.58 – Dynamo script