

Article

Development of a BIM Platform for the Design of Single-Story Steel Structure Factories

Dejiang Wang *  and Haojie Lu

School of Mechanics and Engineering Science, Shanghai University, Shanghai 200444, China; luhaojie@shu.edu.cn
* Correspondence: djwang@shu.edu.cn

Abstract: Traditional design methods for single-story steel structure factories are characterized by low levels of digitalization and high error rates. To deal with these problems, a building information modeling (BIM) platform for the design of single-story steel structure factories was developed in this paper, which aimed to improve the design process for such structures. Firstly, the components of the factory were categorized, and the Revit API was employed to automate the generation of the BIM model. Load applications and combinations were then established using the Revit API, which relied on a set of predefined parameters. Secondly, this paper proposed the creation of a dedicated database for data exchange between BIM software and finite element analysis software. Additionally, the SAP2000 Open Application Programming Interface (OAPI) was employed for the automated construction and analysis of the SAP2000 structural model. Finally, the innovative use of Dynamo–Revit API hybrid programming allowed for the visualization of internal forces directly within the Revit environment, significantly diminishing the dependency on standalone FEA software. The application results obtained on a project demonstrated that the developed platform markedly improves the efficiency of design single-story steel structure factories and ensures the accuracy of the structural analysis. This confirms that the developed platform can transform the traditional design process by integrating advanced digital tools, thereby providing a novel approach to the design of single-story steel structure factories.

Keywords: building information modeling (BIM); single-story steel structure factory; structure calculation; Revit secondary development



Citation: Wang, D.; Lu, H.

Development of a BIM Platform for the Design of Single-Story Steel Structure Factories. *Buildings* **2024**, *14*, 747. <https://doi.org/10.3390/buildings14030747>

Academic Editor: Ahmed Senouci

Received: 28 January 2024

Revised: 22 February 2024

Accepted: 9 March 2024

Published: 10 March 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Single-story steel structure factories, especially those constructed with lightweight portal frames, are widely used in China due to their low cost, reduced weight, and shorter construction periods [1]. However, the prevalent structural design approach for such buildings remains two-dimensional (2D), a practice driven by historical preferences, resulting in final designs typically depicted in planar CAD drawings [2]. This conventional design approach omits a three-dimensional (3D) perspective, potentially increasing the risk of design inaccuracies and subsequent revisions [3]. In addition, the traditional two-dimensional design workflow often segregates disciplines such as architectural, structural, and MEP, leading to poor coordination, thereby impacting the overall quality and efficiency of the finished project.

BIM, as a continuously evolving advanced digital technology and process, has been triggering a sustained revolutionary impact on the architecture, engineering, and construction (AEC) industry [4–6]. Unlike the static 2D drawings, BIM models are dynamic, serving as a relational database that stores all information regarding building objects throughout their lifecycle. This comprehensive repository of data supports informed decision-making throughout the entire project lifecycle, from initial design to final demolition [7].

In the design phase, BIM technology allows designers to transition from traditional 2D design to 3D parametric design. This transition not only enhances the accuracy and

speed of design processes but also fosters improved collaboration and facilitates integrated project delivery [8–10]. Single-story steel structure factories have highly similar structural components and layouts. This uniformity makes them particularly well suited to the application of BIM technology for the generation of 3D models and the automation of design processes.

The application of BIM technology during the design of single-story steel structure factories involves two challenges. Firstly, the existing BIM modeling tools do not adequately support the automatic generation of models for this type of structure, leading to inefficiencies in the modeling process. Secondly, most BIM software possesses excellent 3D modeling and visualization capabilities, but often lacks structural calculation functions [11,12]. Consequently, when applying BIM models to structural calculation and analysis, it is common practice to import the BIM model into professional finite element software via data exchange. Although the Industry Foundation Classes (IFC) format used for data exchange was designed to maintain a comprehensive dataset of BIM information [13], issues with information loss still arise during the data transfer process [14,15]. Designers are compelled to manually correct any missing or erroneous data, which reduces design efficiency and increases labor costs.

Although the previous studies have explored the above two challenges and proposed some solutions to apply BIM technology in the field of steel structures, there is a conspicuous absence of an integrated software platform and a complete design workflow specifically for the design of single-story steel structure factories.

Therefore, the purpose of this study is to develop a BIM platform for the design of single-story steel structure factories. This platform integrates Revit 2018, SQL Server 2019, and SAP2000 v23 software for modeling, data exchange, and structural analysis. Firstly, using the C# programming language and the Revit 2018 API, the platform enabled the automated generation of BIM models, as well as the application and combination of loads. Furthermore, a specialized SQL Server database was developed for data exchange, and the SAP2000 OAPI was utilized for structural calculation and analysis. In addition, the platform enabled the display of internal force results within the Revit environment. By using this platform, designers can perform structural design and optimization more efficiently, ensuring the precision and reliability of their design projects.

2. Literature Review

BIM technology has been widely applied in the field of steel structures, with many researchers developing specialized tools and methods. Wei et al. [16] developed a 3D software system called SteelGate, which was constructed by using the object-oriented design method. It included a parametric database for structural elements such as steel panels and beams, equipped with tools for structural analysis and generation of construction drawings. While highly effective for steel gate design, the software may be less applicable to other structural forms. A web-based platform developed by Chen et al. [17] was specifically engineered for the manufacturing phase of steel structures, addressing common issues such as production delays and quality inconsistencies. However, it is important to note that the platform did not extend its capabilities to the earlier stages of structural design and analysis, which were critical for the overall engineering process. Moreover, Carroll et al. [18] utilized the visual programming language Dynamo to automate repetitive tasks in the steel structure design process, specifically focusing on the generation of BIM and an analysis model. Laefer et al. [19] proposed a method to employ kernel density estimation for determining the shape and section size of steel structure components from point cloud data, which helped to generate geometry in a BIM-compatible format. Yang et al. [20] provided an approach for the retrieval of geometric details from laser-scanned data of steel structures, facilitating the automated development of an as-built BIM model. Basta et al. [21] established an equation model for Steel Structure Deconstructability Assessment Scoring (SS-DAS), and Dynamo was utilized to create a BIM-based framework to assess steel structure deconstructability. Avendaño et al. [22] designed an integrated BIM model

for steel structure construction projects, aimed at providing systematic, efficient, and precise guidance during the construction of steel structures. Yoo et al. [23] compared the efficiency of using 2D drawings versus 3D models during the design and manufacturing stages of steel structures, and they analyzed the advantages of using 3D models quantitatively.

Interoperability in BIM is the capability of different software applications to communicate and exchange data efficiently and accurately [24], which is a critical factor for the application of BIM in structural design. The literature review by Vilutiene et al. [25] demonstrated that there has been a significant increase in the volume of research concerning the application of BIM in structural design since 2014. Nevertheless, when compared to the application of BIM in other domains, the quantity of studies pertinent to structural design remains comparatively low. Additionally, their findings suggest that recent research themes in structural design are intimately associated with interoperability. Shin et al. [26] emphasized the dependence of structural quality on interoperability, which can be realized through openBIM platforms or application-programming interfaces. Musella et al. [27] utilized an openBIM platform based on the standard data format IFC to manage existing structures before and after seismic events. Gerbino et al. [28] examined CAD-CAE interoperability issues and developed mechanisms for bidirectional data exchange between AEC CAD/CAE software to establish a baseline for data exchange. Tang et al. [12] constructed a data conversion interface between BIM software and the finite analysis software ABAQUS. The interface enables BIM modeling data to be interoperable with finite element analysis software to realize structural calculation within BIM. Despite these advancements, the goal of perfectly interoperable integrated systems remains elusive [29]. Researchers continue to seek more efficient implementations of BIM by developing high-performance tools, enhancing interactions, and optimizing processes under different conditions [30].

While some studies have explored the application of BIM in the domain of steel structures and its interoperability within structural design, there remains a lack of a software platform integrating the functions of BIM modeling and structural analysis specifically for single-story steel structure factories.

3. Methodology

In this paper, the design process for single-story steel structure factories was optimized by developing a dedicated BIM platform. This platform integrates Revit 2018, SQL Server 2019, and SAP2000 v23 software, enabling the automatic generation of BIM models, the application of loads, structural calculations, and the visualization of internal forces. The framework of the developed platform is illustrated in Figure 1 and described below:

1. BIM model auto-generation: Leveraging Revit secondary development technology, the platform automatically constructed a BIM model of a single-story steel structure factory based on parameters entered by the designer. This model included all critical structural elements, such as columns, beams, and bracing.
2. Setting calculation parameters: Once the BIM model was constructed, the platform generated appropriate loads and load combinations based on the structural calculation parameters set by the designer.
3. Structural calculation and analysis: The geometric information and structural parameters of the BIM model were transferred to a specialized database. Utilizing the SAP2000 OAPI, the platform automatically created an SAP2000 model and ran the analysis. The analysis results were subsequently stored in the database for review, verification, and as part of the project's documentation.
4. Internal force visualization: The platform utilized the internal force results stored in the database, employing Dynamo–Revit API hybrid programming, to render internal force diagrams within the Revit environment. These diagrams enabled designers to conduct in-depth analysis and verification of the structural elements.

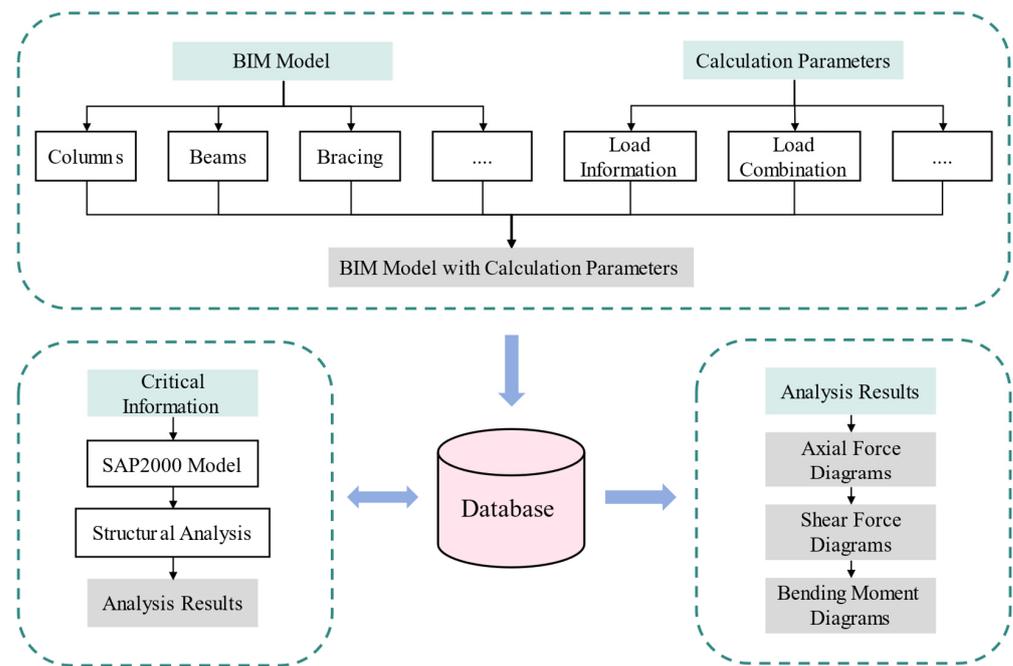


Figure 1. The framework of the developed platform.

3.1. Generation of the BIM Model

The BIM model of a single-story steel structure factory was automatically generated using Revit secondary development technology. This automation was driven by designer-specified parameters, such as bay width, spacings, materials, and section types. The portal frame structure is the most commonly used system for single-story steel structure factories in China. The primary structural components of this type of factory include the main frames, the bracing system, and the anti-wind columns. The target model generated is shown in Figure 2.

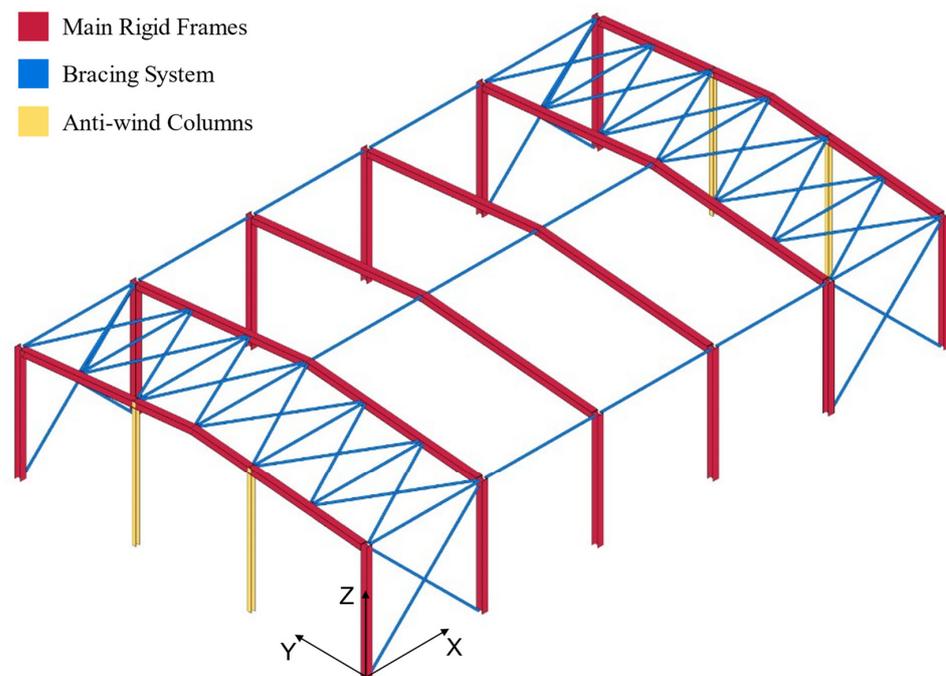


Figure 2. The target BIM model.

3.1.1. Generation of Main Frames

The main frames, consisting of columns and beams, provide the basic load-bearing capacity for the entire structure. The basic approach to automating the generation of columns using the Revit API is as follows:

1. Invoke the `Create.NewFamilyInstance()` method to create column instances based on specified parameters.
2. Access the parameters of the column instances that need to be modified using the `element.get_Parameter()` method, and then adjust these parameters using the `set()` method.

During the column creation process, the key method `Create.NewFamilyInstance()` involves four parameters: location, symbol, level, and `structuralType`. The symbol was obtained based on the designer's selection of the column type or family symbol from the Revit library. The level was set to the base of level in the Revit environment. The `structuralType` was determined by the nature of the components. For columns, this was set to `StructuralType.Column`. Thus, only the location, an XYZ point that defines the physical placement of the column, needed to be calculated based on the bay width and spacings. As depicted in Figure 3, B denotes the bay width of the portal frame, and s_i denotes the spacing for the i -th span of the frame. Two Lists, `colBtmList1` and `colBtmList2`, were utilized to store the base coordinates of the columns on both sides of the portal frame. The index of the List represents the span number of the frame. The count of the List represents the total number of frames. In `colBtmList1` and `colBtmList2`, the Z-coordinate values for all location points were set to 0. For location points in `colBtmList1`, the Y-coordinate values were set to 0; for those in `colBtmList2`, the Y-coordinate values were set to B . The X-coordinate values in `colBtmList1` and `colBtmList2` were calculated by cumulatively adding the spacings s_i . Following the creation of the initial column model, the material properties and top elevation of the columns were adjusted by design requirements.

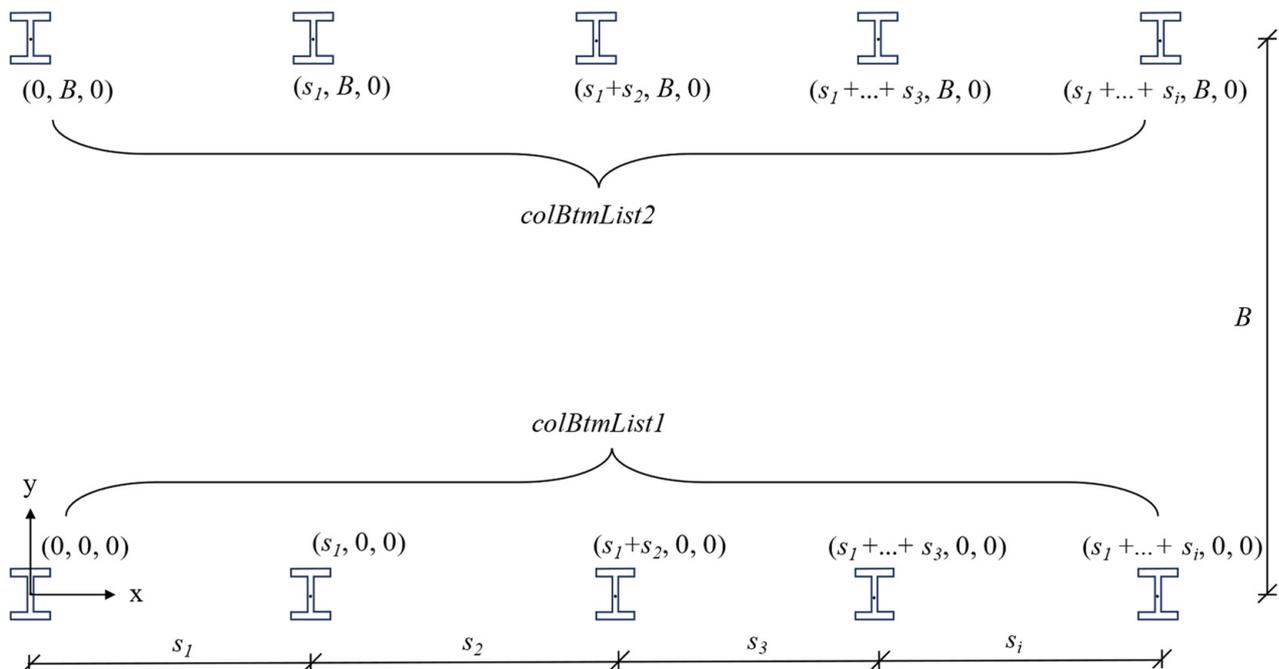


Figure 3. The location points of the column bases.

In contrast to column generation, which was based on a single location point, beam generation relies on a location curve. The steps for beam generation using Revit API are as follows:

1. Calculate the start and end coordinates of the beam by vector addition, using the column height and the roof height input by the designer.
2. Use the `Line.CreateBound()` method to create a location line based on the calculated coordinates of the two endpoints.
3. Invoke the `Create.NewFamilyInstance()` method, passing the established location line as a parameter to generate the beam instance.
4. Adjust the beam instance parameters, using the `element.get_Parameter()` method followed by the `set()` method for necessary modifications.

As demonstrated in Figure 4, $colBtmList[i]$ ($colBtmList1[i]$ and $colBtmList2[i]$) represents the base coordinate of the column in the i -th span frame, h_{col} is the column's height, and h_r is the height of the roof. The two endpoints for $Line1[i]$ and $Line2[i]$ can be calculated as follows:

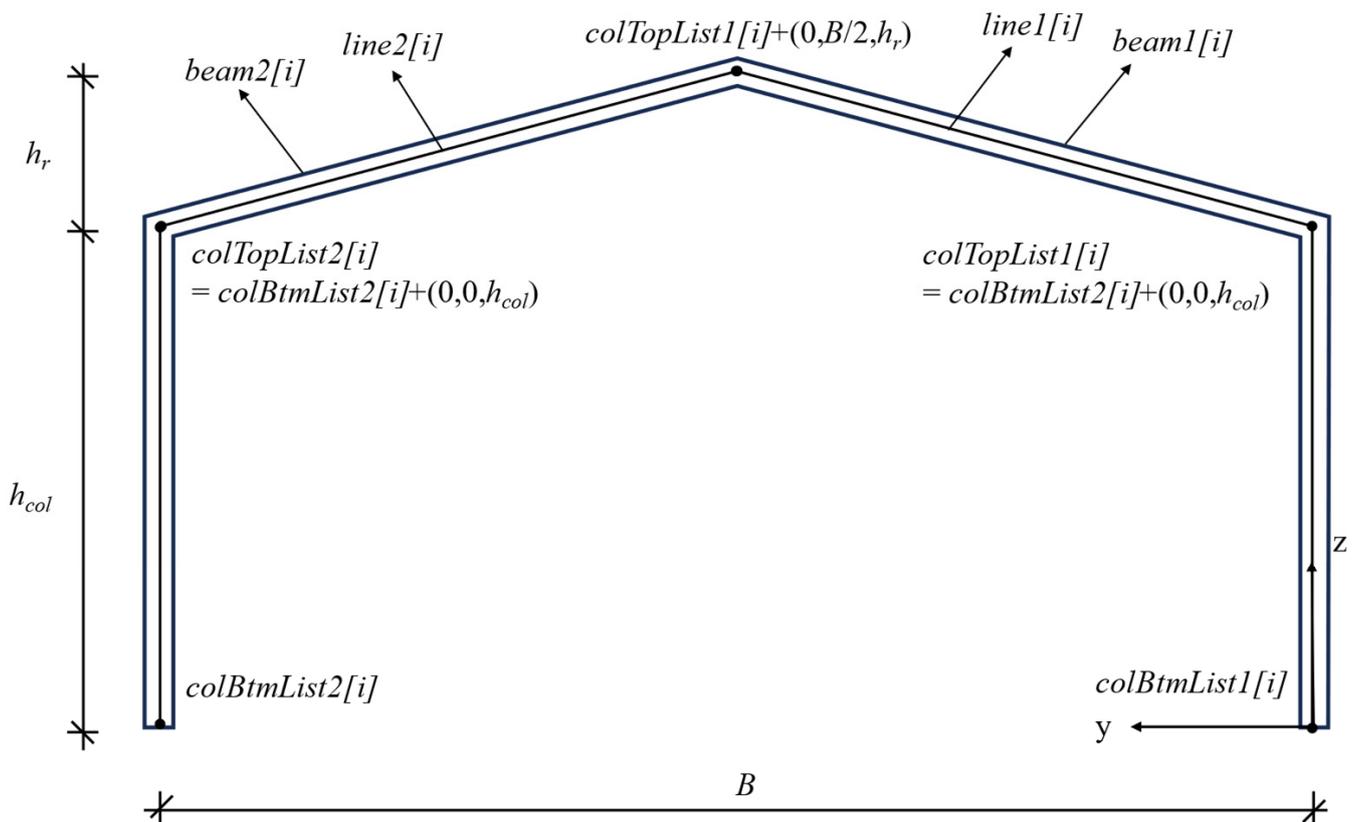


Figure 4. The coordinates for the beams in the i -th span frame.

Start point of $Line1[i]$:

$$colTopList1[i] = colBtmList1[i] + (0, 0, h_{col}) \quad (1)$$

Endpoint of $Line1[i]$:

$$colTopList1[i] + (0, B/2, h_r) \quad (2)$$

Start point of $Line2[i]$:

$$colTopList2[i] = colBtmList2[i] + (0, 0, h_{col}) \quad (3)$$

Endpoint of $Line2[i]$:

$$colTopList1[i] + (0, B/2, h_{col}) \quad (4)$$

Based on the endpoint coordinates calculated above, we used `Line.CreateBound()` to generate the location lines that define the beam axes, and then passed these lines as parameters to the `Create.NewFamilyInstance()` method to generate the beam instances in Revit. Similar to the generation of columns, after the beam instances were generated, the material properties of the beams also needed to be adjusted.

3.1.2. Generation of Anti-Wind Columns

The procedure for generating anti-wind columns was similar to the process for columns detailed in Section 3.1.1. The difference for anti-wind columns lay in an additional step—rotating the columns by 90 degrees within the XY plane to ensure their cross-sections were appropriately oriented to resist wind loads. In the Revit API, the `ElementTransformUtils` class offers the `RotateElement()` method, which facilitates the rotation of elements. As depicted in Figure 5, the rotation axis was a line object that was perpendicular to the XY plane and passed through the midpoint of the column, extending in the direction of the Z-axis. It is important to note that the rotation angle must be expressed in radians, with 90 degrees being equivalent to $\pi/2$ radians.

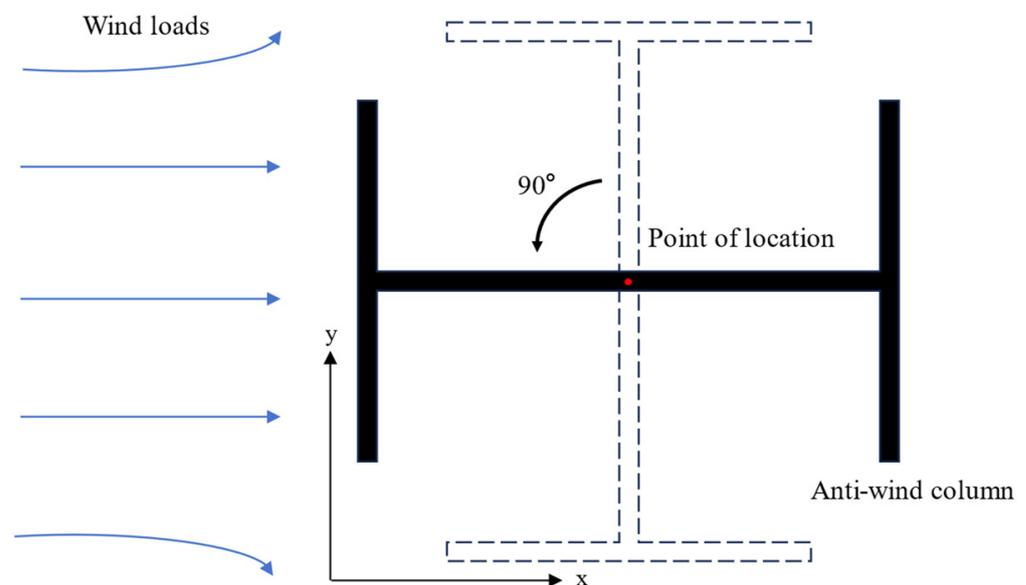


Figure 5. The schematic diagram of anti-wind column rotation.

3.1.3. Generation of Bracing System

The bracing system includes tie rods, column bracing, and roof bracing. Tie rods are typically arranged along the full length of the structure at the top of the columns and at the ridge. Column bracing and roof bracing are commonly positioned within the same span, and they are usually implemented in a cross-bracing pattern to enhance structural stability.

The generation of bracing elements followed a methodology similar to that described for beams in Section 3.1.1. The critical aspect was accurately acquiring the coordinates at both ends of the bracing elements. For tie rods, the end coordinates were derived from the column top or ridge coordinates of the adjacent frames. For column bracing, the end coordinates were determined by the column bottom coordinates and the column top coordinates of the adjacent frame. For roof bracing, it was imperative to perform calculations based on the specified number of roof braces to ensure an even distribution. Algorithm 1 provides the essential code for calculating the end coordinates of the roof bracing and generating the corresponding model. Figure 6 shows the generated bracing system in Revit 2018.

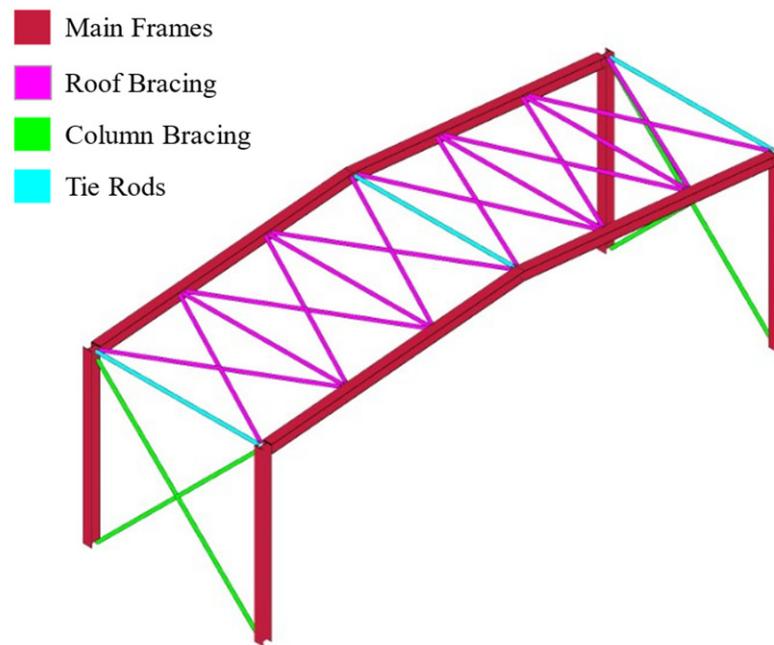


Figure 6. The generated bracing BIM model.

Through the methodology presented above, the developed platform can rapidly construct the BIM model of a single-story steel structure factory as shown in Figure 2. It is a shift from manual to automated modeling, significantly improving modeling efficiency.

Algorithm 1: Generate the roof bracing using the Revit API

Require: *colTopList* containing column top coordinates, *rofTopList* containing roof top coordinates, *n* represents number of roof bracings, *symbol* represents FamilySymbol in the Revit API, *level* represents Reference Level in the Revit API.

```

1  if need to generate roof bracing in the i-th span then
2      for j = 0 to n do
3          start ← colTopList[i] + j/n*(rofTopList[i] − colTopList[i]);
4          end ← colTopList[i + 1] + (j + 1)/n*(rofTopList[i + 1] − colTopList[i + 1]);
5          line1 ← Line.CreateBound(start, end);
6          doc.Create.NewFamilyInstance(line1, symbol, level, StructuralType.Bracings);
7      end for
8      for j = 0 to n − 1 do
9          start ← colTopList[i + 1] + j/n*(rofTopList[i + 1] − colTopList[i + 1]);
10         end ← colTopList[i] + (j + 1)/n*(rofTopList[i] − colTopList[i]);
11         line2 ← Line.CreateBound(start, end);
12         doc.Create.NewFamilyInstance(line2, symbol, level, StructuralType.Bracings);
13     end for
14 end if

```

3.2. Generation of Load Case, Load, and Load Combination

Based on the generated BIM model, load cases, loads, and load combinations were generated by using the Revit API, in accordance with the Chinese load standards.

3.2.1. Generation of Load Case

When designing a single-story steel structure factory, the load cases typically include dead load case, live load case, wind load case, seismic load case, and others, depending on the specific project. These load cases could be defined by utilizing the `LoadCase.Create()` method. Once these load cases were defined, the designer could view and manage them through the Load Cases tab in Revit 2018, as shown in Figure 7.

	Name	Case Number	Nature	Category
1	Dead	1	Dead	Dead Loads
2	Live	2	Live	Live Loads
3	WY+(+i)	3	Wind	Wind Loads
4	WY+(-i)	4	Wind	Wind Loads
5	WY-(+i)	5	Wind	Wind Loads
6	WY-(-i)	6	Wind	Wind Loads
7	WX+(+i)	7	Wind	Wind Loads
8	WX+(-i)	8	Wind	Wind Loads
9	WX-(+i)	9	Wind	Wind Loads
10	WX-(-i)	10	Wind	Wind Loads
11	EX	11	Seismic	Seismic Loads
12	EY	12	Seismic	Seismic Loads

Figure 7. The Load Cases tab in Revit 2018.

3.2.2. Generation of Load

In order to accurately apply loads in the Revit analysis model, it was crucial to convert area loads into equivalent line loads. The equivalence formulas for dead loads and live loads are as follows:

$$G_k = g_k \times b \quad (5)$$

$$Q_k = q_k \times b \quad (6)$$

where g_k denotes the nominal value of dead load (kN/m^2), q_k denotes the nominal value of live load (kN/m^2), and b denotes the loaded width (m).

The equivalence formula for wind loads is as follows:

$$W_k = w_k \times b = 1.1 \times \mu_w \times \mu_z \times w_0 \times b \quad (7)$$

where w_k denotes the nominal value of wind load (kN/m^2), w_0 denotes the basic wind pressure (kN/m^2), μ_w denotes the wind load coefficient, and μ_z is the coefficient for variation of wind pressure with height.

The steps to generate the loads are as follows:

1. Use the `LineLoad.Create()` method to create the line loads acting on the main frames.
2. Adjust the magnitude of the line load and the associated load case by the formulas detailed previously.

The critical parameters for seismic analysis, such as seismic intensity, characteristic period, and damping ratio, were input by the designers into the platform interface. These parameters were then stored in the database.

3.2.3. Generation of Load Combination

Load combinations typically include both basic and seismic load combinations. Basic load combinations comprise various mixes of dead, live, and wind loads, while seismic load combinations are designed to specifically consider the effects of seismic response on the structure.

Using the Revit API to generate load combinations requires following these steps:

1. Instantiate `LoadComponent` objects, initializing the `loadCaseId` and `factor` properties for each individual load case that will be included in the load combination.

2. Invoke the LoadCombination.Create() method to create a new load combination.
3. Utilize the SetComponents() method to set loadComponents assign an array of the previously instantiated LoadComponent objects to the newly created load combination.

Since different standards and countries have different load combination factors, the designer must input these load combination factors based on specific design criteria. The user interface that facilitates the input of these factors and the presentation of the corresponding results is illustrated in Figure 8.

The screenshot shows the Revit software interface with the 'Combination' dialog box open. The dialog is divided into two main sections: 'Factor Setting: Input Factors' and 'Results of Combination: Output Results'. The 'Input Factors' section contains several input fields with numerical values: Structural importance factor (1.0), Partial factor for DL (1.3), Partial factor for LL (1.5), Combination factor for LL (0.7), Partial factor for WL (1.5), Combination factor for WL (0.6), Partial factor for GL (1.3), and Partial factor for SL (1.4). The 'Output Results' section displays a list of 12 load combination formulas, such as '1.3 * Dead + 1.5 * Live + 1.05 * WY+(+i)'. A blue button at the bottom is labeled 'Generate Load Combinations'.

Figure 8. The user interface for inputting load combination factors and displaying results.

3.3. Data Exchange and Structural Analysis

The main advantages of BIM technology lie in its 3D visualization, parametric modeling, and data integration capabilities. However, it falls short in providing adequate capacity for structural calculation and analysis. Some tools within BIM software offer limited structural calculation functions, but compared to professional finite element software, their analytical capabilities and flexibility are constrained [12]. This limitation prevents their widespread adoption among designers. The IFC format used for data exchange contains comprehensive information about the model, often resulting in large file sizes. In contrast, finite element software demands only critical parameters for creating a structural model [14]. Therefore, this paper proposed a method that employs an intermediary database for data exchange between BIM software and finite element software. This approach efficiently transferred model and structural analysis results, addressing the issue of information isolation while also reducing unnecessary information redundancy.

3.3.1. Method of Data Exchange

As shown in Figure 9, this study utilized an SQL Server database to store and manage data in the process of data exchange. The specific steps of data exchange are as follows:

Step 1: Analyze and ascertain the key parameters needed for the SAP2000 model construction and structural analysis, such as the coordinates of each structural member, material properties, and cross-section shapes. Use the Revit API to acquire these parameters and store them into the database.

Step 2: Retrieve the key parameters from the database and utilize the SAP2000 OAPI to construct the SAP2000 model, followed by the automatic initiation of the structural analysis.

Step 3: Store the structural analysis results into the database, including modal periods, member forces, and joint displacements.

Step 4: Extract the structural analysis results from the database and display them through the interface of the developed platform.

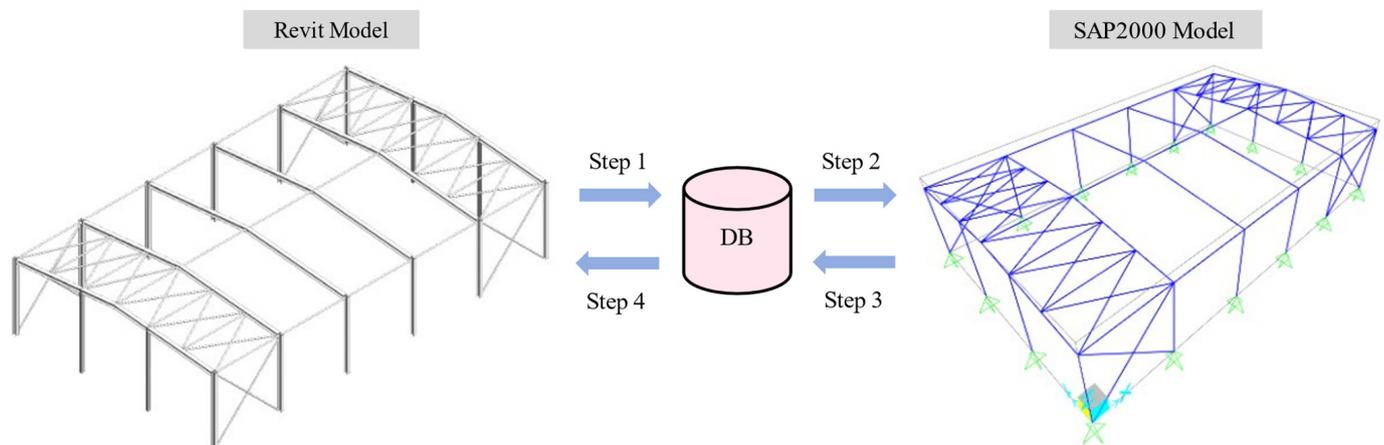


Figure 9. The schematic diagram of data exchange.

3.3.2. Development of Database

The developed platform utilized an SQL Server database for the storage and management of exchange data. SQL Server 2019 is a comprehensive relational database management system (RDBMS) developed by Microsoft, designed to cater to a diverse range of applications. It has been extensively employed in various engineering fields [31]. In addition, ADO.NET technology facilitates connecting, accessing, and manipulating an SQL Server database [32]. The storage structure of the database developed in this study is shown in Table 1.

Table 1. The storage structure of the SQL Server database.

Tables	Fields
Column	Id, Type, Start_X, Start_Y, Start_Z, End_X, End_Y, End_Z, SectionShape, Material, ...
Beam	Id, Type, Start_X, Start_Y, Start_Z, End_X, End_Y, End_Z, SectionShape, Material, ...
Bracing	Id, Type, Start_X, Start_Y, Start_Z, End_X, End_Y, End_Z, SectionShape, Material, ...
Load	Id, Dead_X, Dead_Y, Dead_Z, Live_X, Live_Y, Live_Z, ...
ModalPeriod	Id, LoadCase, StepNum, Period, Frequency, CircFreq, EigenValue
ElementForce	Id, Station, LoadCase, P, V2, V3, T, M2, M3
JointDispl	Id, Joint, LoadCase, U1, U2, U3, R1, R2, R3

3.3.3. Structural Analysis Using SAP2000 OAPI

SAP2000 v23 provides open application programming interfaces that enable developers to automate the model construction and analysis of structural model, as well as the extraction of analysis results [33,34]. In this study, the SAP2000 OAPI was utilized to construct the corresponding SAP2000 model and perform structural analysis, leveraging data exported from Revit 2018 software and stored in the database.

As shown in Figure 10, the process began with the initialization of the SAP2000 application using the `ApplicationStart()` method, followed by the creation of a new SAP2000 model with the `NewBlank()` method. Afterwards, material properties and section attributes were added using the `AddMaterial()` and `SetSection()` methods, respectively. The basic construction of the SAP2000 model was then completed by defining the coordinates of each component from the database using the `FrameObj.AddByCoord()` method, while supports and node constraints required further adjustments. Subsequently, load patterns, load cases, and load combinations were established, and specified loads were applied to the SAP2000 model. For seismic analysis, the response spectrum function was defined using the `FuncRs.SetChinese2010()` method, where the damping ratio, modal combination method, and other parameters were set. Finally, the `RunAnalysis()` method was executed to perform the structural analysis, generating the structural analysis results in SAP2000.

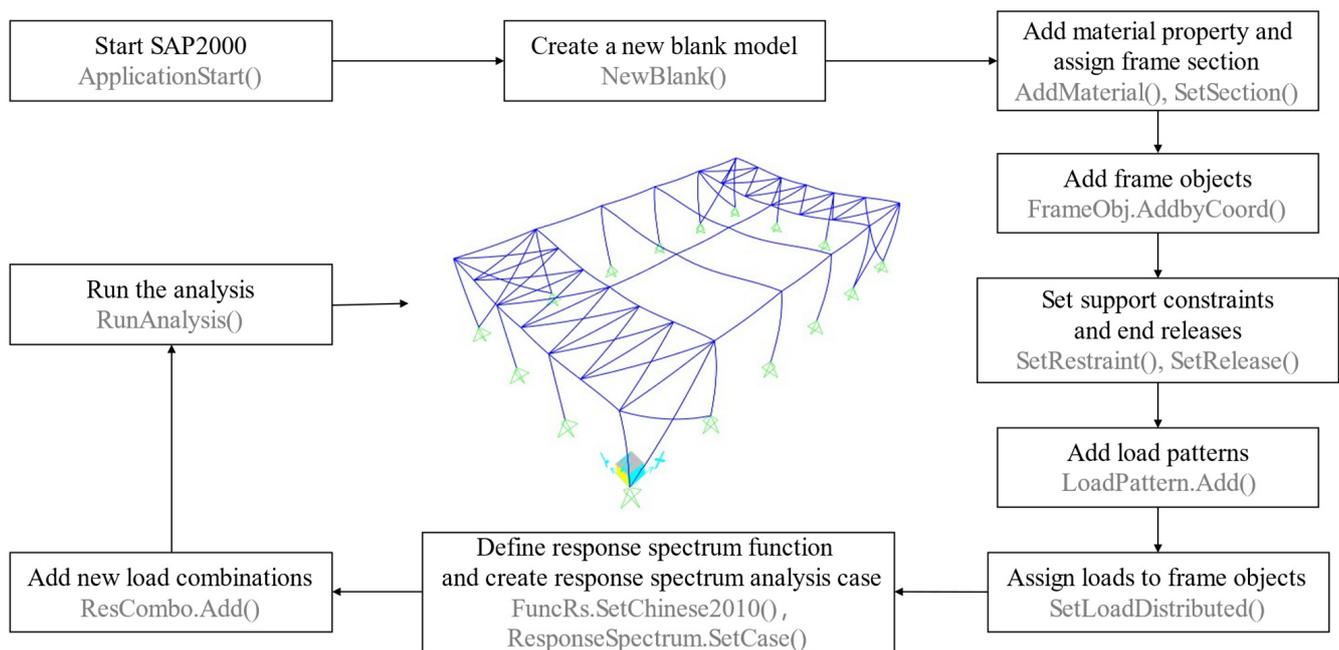


Figure 10. The flowchart of constructing the SAP2000 model and structural analysis.

3.3.4. Analysis Results Export

In order to facilitate designers in querying structural analysis results and reusing them, it was essential to extract the analysis outcomes using the SAP2000 OAPI and store them in the database. The `Result.ModalPeriod()` method can be utilized to acquire the modal periods, frequencies, and other dynamic characteristics. The `Result.FrameForce()` method can be used to obtain the internal forces of members under specific load cases. Additionally, the `Result.JointDispl()` method is available to acquire the displacements of joints under specific load cases. In the platform developed in this paper, the designers can query the structural analysis results on the platform interface.

3.4. Visualization of Internal Force in Revit

To streamline the workflow for designers and enhance the visualization of analysis results, this paper introduced a solution that employs Dynamo–Revit API hybrid programming to draw the internal force diagrams under different load cases in Revit software. By integrating this functionality, designers can efficiently access and view the internal force results within Revit software without repeatedly switching between SAP2000, significantly saving time and improving the efficiency of the design process.

3.4.1. Dynamo–Revit API Hybrid Programming

Internal force diagrams of structural members are typically represented by spatial curves, derived by interpolating a series of control points. Solely depending on Revit API to generate these spatial curves can be a complex and labor-intensive endeavor. To streamline this task, this study employed Dynamo–Revit API hybrid programming, which leverages the capabilities of both Dynamo and the Revit API.

Dynamo is a parametric design platform that operates within the Revit environment, supporting visual programming and featuring a powerful library of geometric nodes [35,36]. The geometric node library in Dynamo, with its diverse array of graphical logic functions, effectively compensates for the shortcomings of Revit software in constructing complex spatial geometries.

In this paper, spatial curves that make up internal force diagrams were generated by invoking functions from Dynamo’s geometric node library. The process used a hybrid programming approach and avoided direct manipulation of the Dynamo interface. The first step was to include the necessary dynamic link libraries in the development environment, specifically *DynamoRevitDS.dll*, *DynamoServices.dll*, *LibG.Interface.dll*, *ProtoGeometry.dll*, and *RevitNodes.dll*. Subsequently, functions from the Dynamo geometric node library were employed to process control point data and generate the required spatial curves. After generating the spatial curves, due to the differences in the representation of geometric objects between the Revit API and Dynamo, the *ToRevitType()* method was utilized to convert Dynamo geometric objects into types recognizable by the Revit API.

3.4.2. Drawing Internal Force Diagrams for Structural Components in Revit

To create internal force diagrams in Revit 2018, the process began by querying the internal force values for each structural member, such as axial force, shear force, and bending moment. These values could be obtained from the database. Then, the offsets for the control points were calculated based on the internal force values and the geometric properties of the member, ensuring that the offsets were proportional to the forces to represent the internal force distribution accurately. Next, smooth curves that represented the internal forces were generated using the *NurbsCurve.ByPoints()* method. Subsequently, these curves were converted into curve objects in Revit API by using the *ToRevitType()* method. Finally, these curve objects were passed as parameters to the *Create.NewModelCurve()* method to draw the internal force diagrams in the Revit environment. To indicate the magnitude of the internal forces, numerical or textual annotations were added to the diagram, ensuring readability. This approach minimizes the need for designers to repeatedly use SAP2000, thereby reducing dependency on finite element analysis software and enhancing overall design efficiency.

4. Case Study

To verify the capabilities and performance of the platform developed in this paper, we conducted a case study of a single-story steel structure factory.

4.1. BIM Modeling and Structural Analysis Processing

The factory is located in Shanghai, covering an area of 756 m² and consisting of eight frames. Both BIM modeling and structural calculations were executed using the platform developed in this paper. The specific steps of using the platform are as follows:

Step 1: Load the platform into the Revit tab and access the platform interface.

Step 2: Enter parameters such as bay width, spacings, and cross-section shapes, then click the ‘Generate’ button to automatically create the BIM model of the single-story steel structure factory. The generated BIM model is shown in Figure 11.

Step 3: Set the load parameters and click the ‘Apply Load’ button to generate various types of loads. The generated loads and the inputted seismic parameters are illustrated in Figure 12. Next, enter the load combination factors and click the ‘OK’ button to generate load combinations.

Step 4: Specify the save path for the SAP2000 file, click the 'Run the Analysis' button, and the platform automatically creates the corresponding SAP2000 model for calculation and analysis, returning the results upon completion. The element forces are demonstrated in Figure 13.

Step 5: Select the load case in the Revit environment to perform internal force checks. The bending moment diagram under the dead load case within the Revit environment is shown in Figure 14.

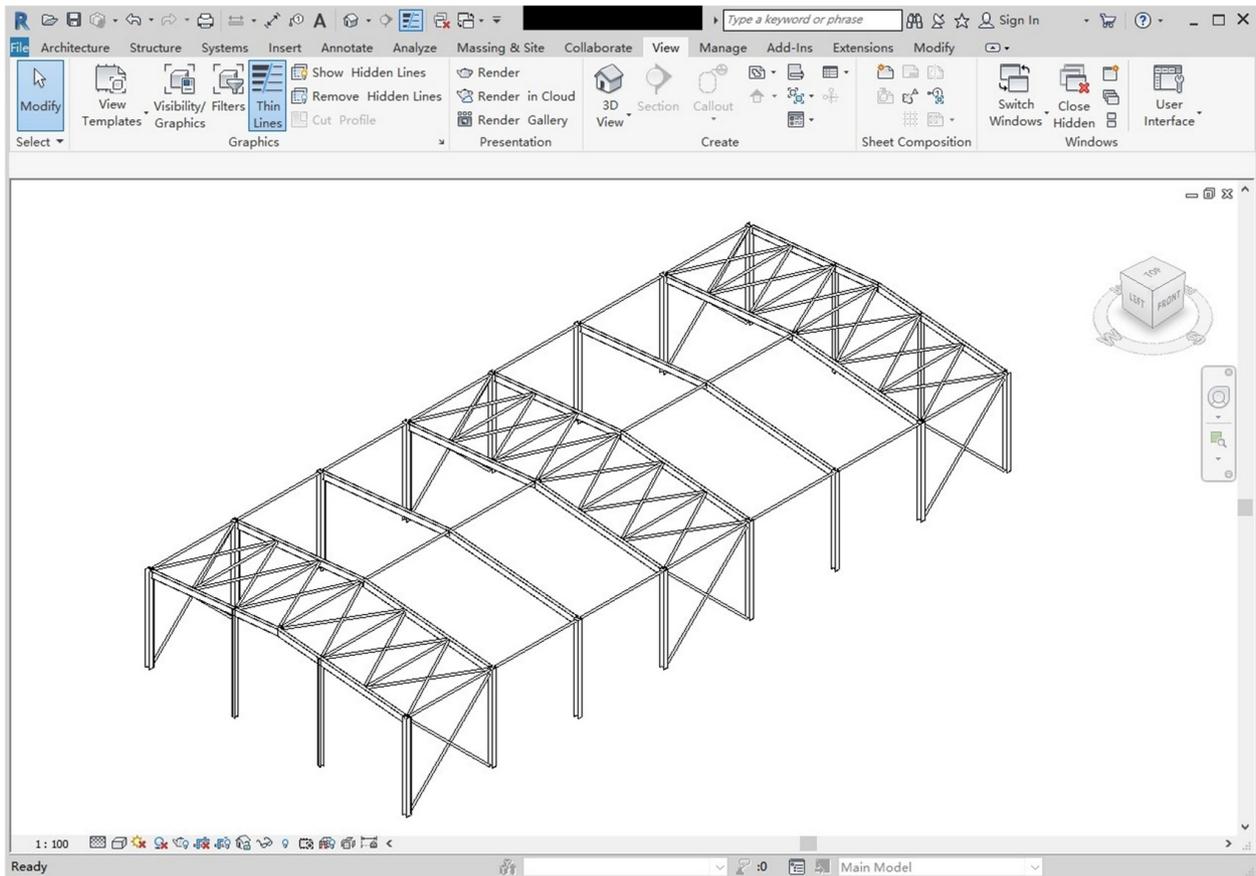


Figure 11. The BIM model generated during the case study.

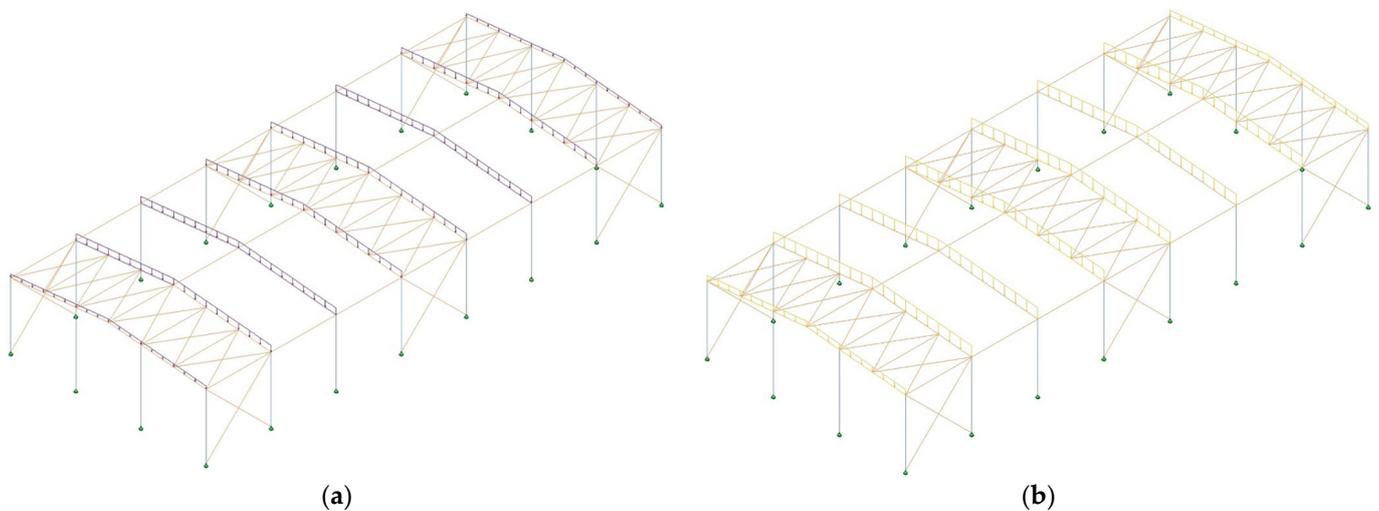


Figure 12. Cont.

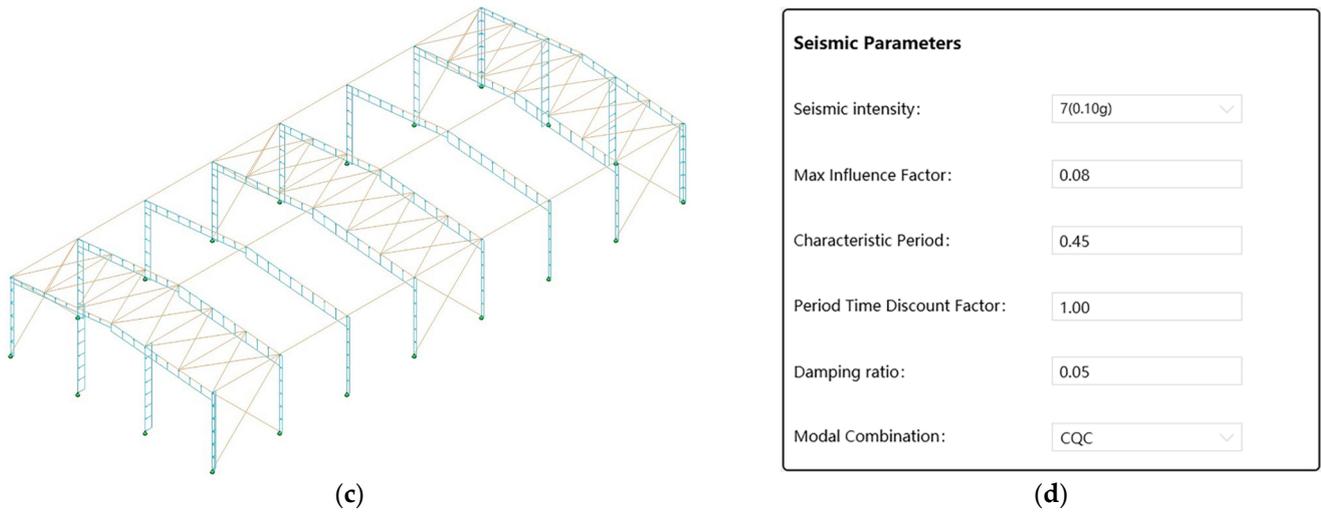


Figure 12. The generated loads and the inputted seismic parameters: (a) the generated dead load; (b) the generated live load; (c) the generated wind load; (d) the inputted seismic parameters.

Revit

Path: C:\Users\15389\Desktop\CaseStudy Folder FileName: CaseStudy .sdb

Run the Analysis

Element Force	Id	Station(m)	P(kN)	V2(kN)	V3(kN)	T(kN-M)	M2(kN-M)	M3(kN-M)
Dead	512264	0	-29.622	8.637	0	0	0	0
Live	512264	3	-26.886	8.637	0	0	-0.001	-25.911
WX+(+i)	512264	6	-24.15	8.637	0	0	-0.002	-51.822
WX+(-i)	512266	0	-30.586	10.564	0	0	0	0
WX-(+i)	512266	3	-27.85	10.564	0	0	-0.001	-31.693
WX-(-i)	512266	6	-25.114	10.564	0	0	-0.001	-63.387
WY+(+i)	512268	0	-29.632	8.952	0	0	0	0
WY+(-i)	512268	3	-26.896	8.952	0	0	0	-26.855
WY-(+i)	512268	6	-24.16	8.952	0	0	-0.001	-53.709
WY-(-i)	512270	0	-34.507	11.277	0	0	0	0
EX	512270	3	-31.771	11.277	0	0	0	-33.831
EY	512270	6	-29.034	11.277	0	0	-0.001	-67.661

Figure 13. The element forces under the dead load case.

Table 3. Comparison of internal force results.

Element Id		Dead Load			Live Load		
		Platform Results	3D3S Results	Discrepancy	Platform Results	3D3S Results	Discrepancy
512280	N (kN)	24.15	24.10	0.2%	18.52	18.46	0.3%
	V (kN)	8.64	8.59	0.6%	6.36	6.32	0.6%
	M (kN·m)	51.82	51.56	0.5%	38.13	37.91	0.6%
512284	N (kN)	24.16	24.16	0.0%	27.14	27.14	0.0%
	V (kN)	8.95	8.92	0.3%	10.5	10.46	0.4%
	M (kN·m)	53.71	53.49	0.4%	63.01	62.76	0.4%
512286	N (kN)	29.03	29.03	0.0%	25.7	25.69	0.0%
	V (kN)	11.28	11.23	0.4%	10.5	10.46	0.4%
	M (kN·m)	67.66	67.381	0.4%	63.01	62.76	0.4%

5. Conclusions

In this study, a BIM platform for the design of single-story steel structure factories was developed using Revit API, SAP2000 OAPI, and other technologies. The platform integrated the BIM modeling function of Revit software and the structural calculation function of SAP2000 software, which provided a new solution for the design of single-story steel structure factories. The specific conclusions are as follows:

1. The platform developed in this paper facilitates the automatic generation of BIM models, the application of loads, structural analysis, and the visualization of internal forces for single-story steel structure factories. A case study was successfully conducted using this platform, effectively demonstrating the entire process from BIM modeling to structural analysis. Furthermore, due to the geometric and design similarities, it is also applicable to other steel structures such as warehouses and buildings, thus contributing to the existing body of knowledge.
2. The platform developed in this paper significantly enhances the efficiency of designing single-story steel structure factories. Compared to traditional methods, the time required for BIM modeling and structural analysis is substantially reduced. Additionally, the platform notably decreases the learning cost of software for designers and simplifies operational complexity.
3. The accuracy of the analysis results obtained through the platform developed in this paper has been thoroughly validated. When compared with results from other calculation software, the maximum deviation is within 0.6%, which satisfies the precision needs of designers.

Although the platform developed in this paper dramatically optimizes the design process of single-story steel structure factories, there are still some limitations. The platform currently includes exclusive integration with SAP2000 for automatic structural analysis and calculation, with no interface for other finite element software. In the future, efforts will be focused on expanding the platform's compatibility to include interfaces for other finite element analysis software.

Author Contributions: Conceptualization, D.W.; methodology, D.W. and H.L.; software, H.L.; validation, H.L.; formal analysis, H.L.; investigation, H.L.; resources, D.W.; writing—original draft preparation, H.L.; writing—review and editing, D.W.; visualization, H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Hu, J. Rational Design of Steel Structure Portal Frame. In Proceedings of the 2020 6th International Conference on Mechanical Engineering and Automation Science (ICMEAS), Moscow, Russia, 29–31 October 2020; pp. 212–216.
2. Czmoch, I.; Pękala, A. Traditional Design versus BIM Based Design. *Procedia Eng.* **2014**, *91*, 210–215. [[CrossRef](#)]
3. Han, S.; Love, P.; Peña-Mora, F. A system dynamics model for assessing the impacts of design errors in construction projects. *Math. Comput. Model.* **2013**, *57*, 2044–2053. [[CrossRef](#)]
4. Alasmari, E.; Martinez-Vazquez, P.; Baniotopoulos, C. An Analysis of the Qualitative Impacts of Building Information Modelling (BIM) on Life Cycle Cost (LCC): A Qualitative Case Study of the KSA. *Buildings* **2023**, *13*, 2071. [[CrossRef](#)]
5. Heaton, J.; Parlikad, A.K.; Schooling, J. Design and development of BIM models to support operations and maintenance. *Comput. Ind.* **2019**, *111*, 172–186. [[CrossRef](#)]
6. Muhammad, M.Z.; Mustapa, M. A Review of Facilities Information Requirements towards Enhancing Building Information Modelling (BIM) for Facilities Management (FM). *IOP Conf. Ser. Mater. Sci. Eng.* **2020**, *884*, 012033. [[CrossRef](#)]
7. Borkowski, A.S. Evolution of BIM: Epistemology, genesis and division into periods. *J. Inf. Technol. Constr.* **2023**, *28*, 646–661. [[CrossRef](#)]
8. Azevedo, G.F. BIM methodology implementation in structural design: Adaptation of procedures and information management. *J. Civ. Eng. Environ. Sci.* **2022**, *8*, 093–099.
9. Hamidavi, T.; Abrishami, S.; Hosseini, M.R. Towards intelligent structural design of buildings: A BIM-based solution. *J. Build. Eng.* **2020**, *32*, 101685. [[CrossRef](#)]
10. Sampaio, Z.; Azevedo, V. Structural Design Developed in a BIM Environment: Benefits and Limitations. *Int. J. Civ. Infrastruct.* **2018**, *1*, 30–41. [[CrossRef](#)]
11. Sami, A.; Abd, A.M.; Mahmood, M. Adopting BIM at Design Phase for Structural Buildings. *Diyala J. Eng. Sci.* **2021**, *14*, 23–35. [[CrossRef](#)]
12. Tang, F.; Ma, T.; Guan, Y.; Zhang, Z. Parametric modeling and structure verification of asphalt pavement based on BIM-ABAQUS. *Autom. Constr.* **2020**, *111*, 103066. [[CrossRef](#)]
13. Yu, Y.; Kim, S.; Jeon, H.; Koo, B. A Systematic Review of the Trends and Advances in IFC Schema Extensions for BIM Interoperability. *Appl. Sci.* **2023**, *13*, 12560. [[CrossRef](#)]
14. Habte, B.; Guyo, E. Application of BIM for structural engineering: A case study using Revit and customary structural analysis and design software. *J. Inf. Technol. Constr.* **2021**, *26*, 1009–1022. [[CrossRef](#)]
15. Muller, M.F.; Garbers, A.; Esmanioto, F.; Huber, N.; Loures, E.R.; Canciglieri Junior, O. Data interoperability assessment though IFC for BIM in structural design—A five-year gap analysis. *J. Civ. Eng. Manag.* **2017**, *23*, 943–954. [[CrossRef](#)]
16. Wei, L.; Wei, Q.; Sun, K. Development of BIM Technology in Steel Structure Design Software. *Appl. Mech. Mater.* **2014**, *501–504*, 2546–2549. [[CrossRef](#)]
17. Chen, S.; Wu, J.; Shi, J. A BIM Platform for the Manufacture of Prefabricated Steel Structure. *Appl. Sci.* **2020**, *10*, 8038. [[CrossRef](#)]
18. Carroll, S.; Lyne, J.; Otreba, M.; McKenna, T. Integrating Computational Design into Structural Engineering Workflows to Enhance Design Automation. In Proceedings of the CitA 5th BIM Gathering—Construction Innovations for Future Generations, Virtually, 21–23 September 2021; pp. 142–149.
19. Laefer, D.F.; Truong-Hong, L. Toward automatic generation of 3D steel structures for building information modelling. *Autom. Constr.* **2017**, *74*, 66–77. [[CrossRef](#)]
20. Yang, L.; Cheng, J.C.P.; Wang, Q. Semi-automated generation of parametric BIM for steel structures based on terrestrial laser scanning data. *Autom. Constr.* **2020**, *112*, 103037. [[CrossRef](#)]
21. Basta, A.; Serror, M.H.; Marzouk, M. A BIM-based framework for quantitative assessment of steel structure deconstructability. *Autom. Constr.* **2020**, *111*, 103064. [[CrossRef](#)]
22. Avendaño, J.I.; Zlatanova, S.; Domingo, A.; Pérez, P.; Correa, C. Utilization of BIM in Steel Building Projects: A Systematic Literature Review. *Buildings* **2022**, *12*, 713. [[CrossRef](#)]
23. Yoo, M.; Ham, N. Productivity Analysis of Documentation Based on 3D Model in Plant Facility Construction Project. *Appl. Sci.* **2020**, *10*, 1126. [[CrossRef](#)]
24. Jiang, S.; Jiang, L.; Han, Y.; Wu, Z.; Wang, N. OpenBIM: An Enabling Solution for Information Interoperability. *Appl. Sci.* **2019**, *9*, 5358. [[CrossRef](#)]
25. Vilutiene, T.; Kalibatiene, D.; Hosseini, M.R.; Pellicer, E.; Zavadskas, E.K. Building Information Modeling (BIM) for Structural Engineering: A Bibliometric Analysis of the Literature. *Adv. Civ. Eng.* **2019**, *2019*, 5290690. [[CrossRef](#)]
26. Shin, T.-S. Building information modeling (BIM) collaboration from the structural engineering perspective. *Int. J. Steel Struct.* **2017**, *17*, 205–214. [[CrossRef](#)]
27. Musella, C.; Serra, M.; Salzano, A.; Menna, C.; Asprone, D. Open BIM Standards: A Review of the Processes for Managing Existing Structures in the Pre- and Post-Earthquake Phases. *CivilEng* **2020**, *1*, 291–309. [[CrossRef](#)]
28. Gerbino, S.; Cieri, L.; Rainieri, C.; Fabbrocino, G. On BIM Interoperability via the IFC Standard: An Assessment from the Structural Engineering and Design Viewpoint. *Appl. Sci.* **2021**, *11*, 11430. [[CrossRef](#)]
29. Turk, Ž. Interoperability in construction—Mission impossible? *Dev. Built Environ.* **2020**, *4*, 100018. [[CrossRef](#)]
30. Abdirad, H.; Pishdad-Bozorgi, P. Trends of Assessing BIM Implementation in Construction Research. In *Computing in Civil and Building Engineering*; ASCE: Reston, VA, USA, 2014; pp. 496–503. [[CrossRef](#)]

31. Mukherjee, S. SQL Server Development Best Practices. *Int. J. Innov. Res. Comput. Commun. Eng.* **2019**, *8*, 3519–3526.
32. Lin, Y.J. ADO.NET Database Access Technology. *Appl. Mech. Mater.* **2015**, *713*, 2217–2220. [[CrossRef](#)]
33. Sotiropoulos, S.; Lagaros, N.D. Topology optimization of framed structures using SAP2000. *Procedia Manuf.* **2020**, *44*, 68–75. [[CrossRef](#)]
34. Usta, O.; Altintas, G. *Parametric Design and Investigation of Base Isolators Using the Script Developed on SAP2000 OAPI*; Manisa Celal Bayar University: Manisa, Turkey, 2020.
35. Collao, J.; Lozano-Galant, F.; Lozano-Galant, J.A.; Turmo, J. BIM Visual Programming Tools Applications in Infrastructure Projects: A State-of-the-Art Review. *Appl. Sci.* **2021**, *11*, 8343. [[CrossRef](#)]
36. Kossakowski, P.G. Visual Programming as Modern and Effective Structural Design Technology—Analysis of Opportunities, Challenges, and Future Developments Based on the Use of Dynamo. *Appl. Sci.* **2023**, *13*, 9298. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.