Dual-Path Implementation and Application of Parametric Design for Shield Tunnel Segments

Yifan Shi

North China University of Water Resources and Electric Power, Zhengzhou, Henan, China

Abstract

To address the issues of low modeling efficiency and poor data-model associativity in shield tunnel engineering, this study explores two distinct BIM-based parametric modeling paths. The first path utilizes the Revit platform as a foundation, establishing a "data-program-component" collaborative drive mode through the Dynamo visual programming environment. This approach achieves the automated placement of segments under complex spatial poses by leveraging parametric adaptive families. The second path is based on the OpenRail platform, utilizing API-based custom development to encapsulate segment selection algorithms and detailed structural logic into integrated plugins. Comparative results demonstrate that the visual programming path offers high flexibility and logic transparency, making it highly suitable for customized projects with frequent design iterations. In contrast, the API-based plugin path excels in computational stability and standardization, catering to the requirements of large-scale production in mega-projects. The research findings provide a robust reference for achieving forward design and digital delivery in shield tunnel engineering.

Keywords

Shield Tunnel Segment; Parametric Modeling; BIM; API-based Development; Forward Design.

1. Introduction

With the rapid advancement of urban rail transit and cross-basin water diversion projects in China, the shield tunneling method has emerged as the mainstream technique for underground construction, owing to its minimal environmental impact and high degree of automation[1-3]. Driven by the industry paradigms of "intelligent construction" and "construction industrialization," Building Information Modeling (BIM) technology has become a pivotal tool for enhancing engineering quality, facilitating collaborative design and fine-grained management throughout the entire project lifecycle[4-6]. However, as the fundamental structural unit of tunnel linings, shield tunnel segments are characterized by complex geometries and intricate assembly logic. Traditional static modeling approaches often encounter significant bottlenecks when dealing with large-scale alignment adjustments or frequent design changes. These limitations manifest primarily as low modeling efficiency, poor data-model associativity, and a high intensity of repetitive manual tasks.

To address these challenges, parametric modeling technology has facilitated a design paradigm shift from traditional "manual drafting" to "logic-driven generation" by defining geometric constraints and logical rules[7]. Currently, scholars both domestically and internationally have conducted extensive research into parametric development across various BIM platforms. One research direction focuses on integrating BIM platforms with visual programming tools to explore dynamic component placement methods in complex spatial environments[8, 9]; Another direction emphasizes API-based custom development to achieve the encapsulation and integration of specialized plugins[10]. Despite the significant achievements in algorithmic implementation within individual paths, a critical gap remains. Due to discrepancies in logical architectures and development thresholds among different platforms, the industry still lacks a comprehensive comparative analysis regarding the practical performance, modeling precision, and situational applicability of these distinct technical methodologies.

To provide a more valuable reference for technical selection in tunnel BIM forward design, this paper investigates two mainstream parametric modeling paths: first, a collaborative path driven by visual programming and adaptive components; and second, a secondary development path focused on functional integration via underlying APIs. By elucidating the construction workflows, data-driven mechanisms, and engineering application results of both methodologies, their respective advantages, disadvantages, and application boundaries are summarized. The research outcomes aim not only to enhance the modeling efficiency of shield tunnels but also to provide standardized underlying logical support for subsequent digital delivery and the construction of Digital Twin systems.

2. Geometric Characteristics and Parametric Logic of Shield Tunnel Segments

2.1. Geometric Morphology of Shield Tunnel Segments

As the core structural component in shield tunneling, the shield tunnel segment serves as the primary barrier against earth and groundwater pressure (as shown in Figure 1). As the permanent lining of the tunnel, segments are widely utilized across a diverse range of applications, spanning from conventional urban rail transit tunnels and utility corridors to large-diameter river-crossing or sea-crossing highway tunnels, as well as ultra-long-distance water conveyance tunnels. Although the outer diameter and thickness of segments are adjusted according to specific load-bearing requirements in different engineering scenarios, their fundamental geometric composition and structural logic exhibit highly standardized characteristics. This consistency provides a solid theoretical foundation for the implementation of parametric modeling.

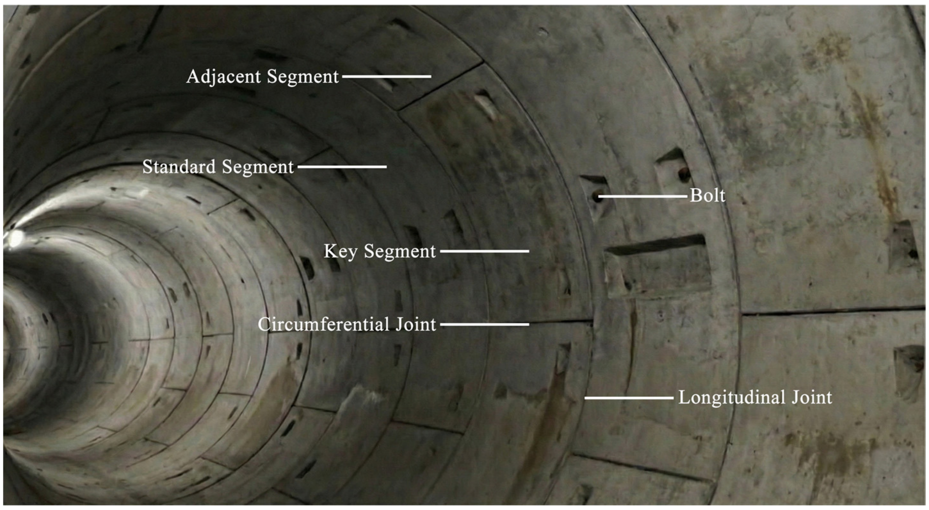


Fig 1. Geometric morphology and structural constituents of a shield tunnel segment

From a structural perspective, a complete shield tunnel segment ring is assembled from several precast concrete units using connectors. Based on differences in assembly sequence and geometric functions, these segments are typically classified into standard segments (A-type), adjacent segments (B-type), and a key segment (K-type). Standard segments occupy the majority of the ring's circumference and possess relatively regular geometries. Adjacent segments are positioned on either side of the key segment, with one side featuring a specific inclination angle to reserve a wedged space for the final insertion of the key segment. The key segment, also known as the closing block, is usually smaller and characterized by significant longitudinal or radial taper angles. It serves as the core component for ensuring ring closure and generating pre-tightening force. In the parametric modeling process, the logical attribution of each segment and the calculation of spatial coordinates for assembly points are critical for model generation.

The refined characteristics of segments are primarily reflected in their intricate detailed structures. The contact surface between adjacent rings is termed the circumferential joint, while the interface between segments within the same ring is the longitudinal joint. The design of sealing gasket grooves within these joints directly dictates the tunnel's waterproofing performance. To accommodate variations in horizontal and vertical curves (i.e., turning or climbing), segment rings are often designed with a specific taper amount, controlling axis deflection by adjusting the width differential between the front and rear faces of the ring. Furthermore, a multitude of components are integrated within the segments, including circumferential and axial bolts for fastening, bolt pockets for operational access, and pre-defined holes for subsequent grouting and hoisting. These detailed features are not only geometrically complex but also dynamically change position according to the segment's division angle. Therefore, in the parametric logic, these geometric constituents must be abstracted into a series of interconnected variables, enabling the automated construction of high-precision shield tunnel models through constraint-driven mechanisms.

2.2. Parametric Design Workflow

2.2.1. Analysis of the Core Advantages of Parametric Modeling

Compared with traditional modeling approaches based on static geometric composition, the core value of parametric modeling lies in its facilitation of a design paradigm shift from "result drafting" to "logic-driven generation." By establishing precise topological relationships and logical constraints, parametric modeling solidifies complex engineering design criteria within the underlying algorithms of the model, achieving a deep coupling between geometric morphology and engineering attributes. When addressing shield tunnel projects characterized by extreme spatial complexity (e.g., overlapping segments of horizontal and vertical curves), traditional modeling often struggles with the cumbersome process of accurately capturing the instantaneous pose of the tunnel axis in 3D space. In contrast, parametric methods leverage spatial analytical geometry algorithms to calculate the tangent, normal, and binormal directions of segment rings within a 3D mileage coordinate system in real-time, ensuring absolute precision in spatial positioning.

Furthermore, parametric modeling establishes a robust linkage mechanism between the "data source" and the "model." This linkage is manifested not only in the automated updating of models and rapid generation of drawings following design changes but also in the efficiency and accuracy of processing large-scale repetitive components. Through parametric means, designers can transform engineering expertise into algorithmic models. During the selection of numerous design schemes, computational design can be employed to perform multi-objective optimization on key variables such as segment layout and taper distribution, thereby achieving precise control over structural performance and construction feasibility at the early design stage.As the core vehicle for digital delivery, the attribute-rich parametric model provides standardized and structured data support for subsequent Finite Element Method (FEM) mechanical simulations, 4D construction simulations, and the construction of Digital Twins during the smart operation and maintenance (O&M) phase. Its value permeates the entire engineering lifecycle.

2.2.2. Logical Architecture of the Two Technical Paths

The two parametric modeling paths selected for this study represent the representative paradigms of "visual programming-driven" and "underlying API-based integration" in current BIM applications.

The first path focuses on a "data-logic-component" collaborative drive mode. Utilizing the BIM platform as a carrier, its core logic lies in constructing a central control program within a visual programming environment. First, stationing data of the shield alignment is retrieved through an external data interface. The program then performs a coordinate transformation from the alignment centerline to the spatial pose of the segments. Subsequently, programming nodes are employed to precisely capture the constraint control points within predefined parametric adaptive families. By transmitting the calculated spatial vector data to the geometric parameters of the family file in real-time, the program drives the precise positioning and orientation of these adaptive components along complex 3D paths. This methodology significantly lowers the threshold for parametric modeling while offering high flexibility and interactivity. The logical workflow is illustrated in Figure 2.

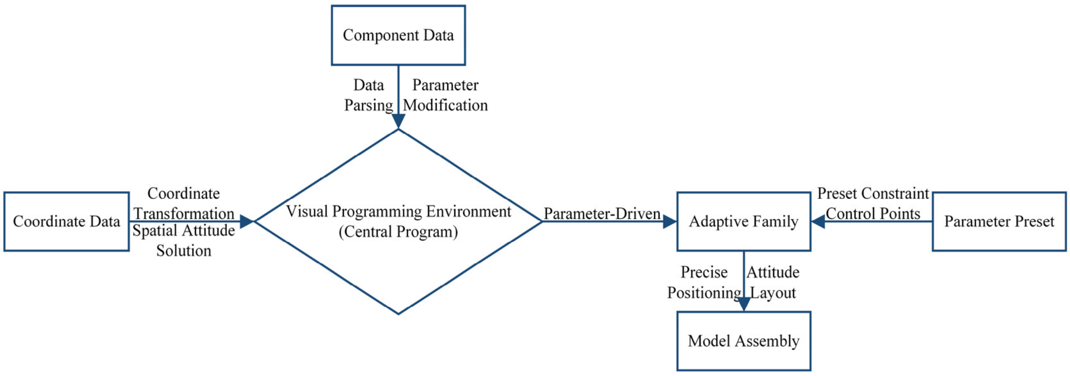


Fig 2. Collaborative "data-logic-component" drive framework via visual programming

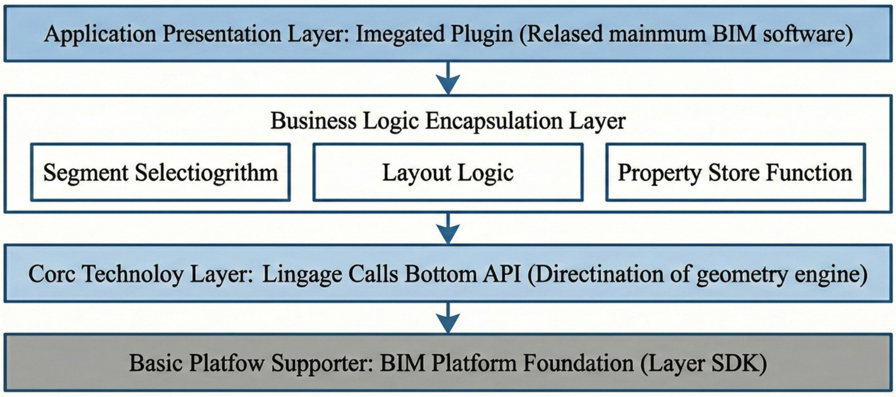


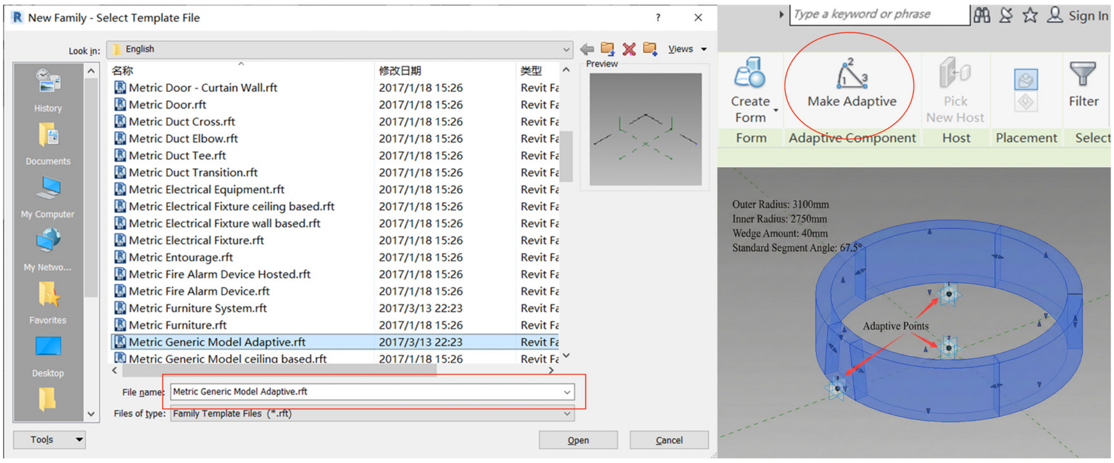
Fig 3. Integrated architecture of the parametric modeling plugin based on underlying APIs

The second path is dedicated to achieving deep functional integration and professional customization through underlying custom development. This methodology, based on the BIM platform, leverages programming languages to invoke the Software Development Kit (SDK), thereby encapsulating complex segment selection algorithms, placement logic, and attribute management functions into standardized software plugins. Unlike visual programming, the API-based path bypasses the combination of general-purpose nodes and instead interacts directly with the geometric engine via underlying APIs. This direct manipulation enables superior computational robustness and execution efficiency when processing data for large-scale segment assemblies. By simplifying cumbersome operational workflows into integrated functional modules and deploying customized plugins within mainstream BIM software, this path achieves a high degree of automation and standardization in the modeling process. The logic of these functional modules is illustrated in Figure 3.

3. Construction of Parametric Models Based on Revit + Dynamo

3.1. Adaptive Family Construction

In the BIM design hierarchy, traditional family files are often constrained by specific floor levels or planar references, making it difficult to accommodate the complex 3D spatial pose variations characteristic of tunnel engineering. To address this, this study utilizes the "Adaptive Generic Model" template as the foundation for segment modeling. The core advantage of adaptive families lies in their liberation from the constraints of traditional family coordinate systems. By defining reference points with topological relationships in 3D space and designating them as "adaptive points," the component can automatically adjust its spatial pose and geometric morphology according to externally input coordinate data. During the construction process, a series of indexed adaptive reference points are placed based on the geometric center and key boundary points of the segment. These points serve not only as positioning benchmarks for component generation but also as logical interfaces for data interaction with subsequent driver programs. The finalized adaptive family is shown in Figure 4.



Look in: English

名称	修改日期	类型
Metric Door - Curtain Wall.rft	2017/1/18 15:26	Revit Fa
Metric Door.rft	2017/1/18 15:26	Revit Fa
Metric Duct Cross.rft	2017/1/18 15:26	Revit Fa
Metric Duct Elbow.rft	2017/1/18 15:26	Revit Fa
Metric Duct Tee.rft	2017/1/18 15:26	Revit Fa
Metric Duct Transition.rft	2017/1/18 15:26	Revit Fa
Metric Electrical Equipment.rft	2017/1/18 15:26	Revit Fa
Metric Electrical Fixture ceiling based.rft	2017/1/18 15:26	Revit Fa
Metric Electrical Fixture wall based.rft	2017/1/18 15:26	Revit Fa
Metric Electrical Fixture.rft	2017/1/18 15:26	Revit Fa
Metric Entourage.rft	2017/1/18 15:26	Revit Fa
Metric Fire Alarm Device Hosted.rft	2017/1/18 15:26	Revit Fa
Metric Fire Alarm Device.rft	2017/1/18 15:26	Revit Fa
Metric Furniture System.rft	2017/3/13 22:23	Revit Fa
Metric Furniture.rft	2017/1/18 15:26	Revit Fa
Metric Generic Model Adaptive.rft	2017/3/13 22:23	Revit Fa
Metric Generic Model ceiling based.rft	2017/1/18 15:26	Revit Fa

File name: Metric Generic Model Adaptive.rft
Files of type: Family Template Files (*.rft)

Outer Radius: 3100mm
Inner Radius: 2750mm
Wedge Amount: 40mm
Standard Segment Angle: 67.5°

Fig 4. Development workflow of the adaptive segment family and configuration of constraint control points

3.2. Visual Programming based on Dynamo

As the geometric benchmark for parametric modeling, the alignment control program fulfills the core function of converting discrete design parameters into a continuous 3D spatial path. This program module (as shown in Figure 5) first retrieves raw data—including stationing, deflection angles, and gradients of the shield alignment—via an external data interface.It utilizes spatial analytical geometry algorithms to establish an alignment model compatible with transition, circular, and vertical curves. By constructing high-order polynomials or integration algorithms within the program, Dynamo generates accurate horizontal alignments and synthesizes them with the amplitude characteristics of vertical curves to produce a spatial axis reflecting the actual tunnel trajectory. Furthermore, the module incorporates a fitting and verification function based on taper amounts. Predicting the deviation between the design axis and the actual assembly centerline of the segment rings, it provides a high-precision positioning reference for subsequent refined layout.

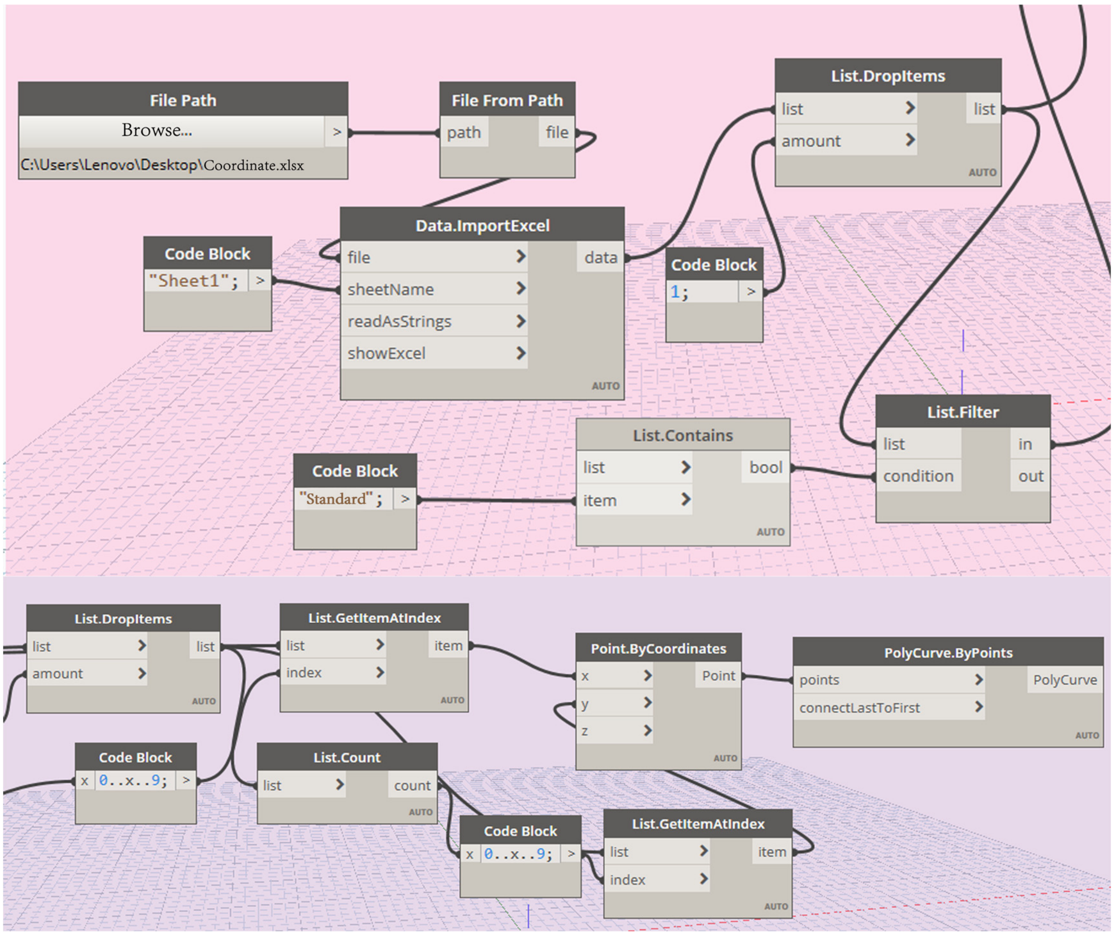


Fig 5. Algorithmic logic of the 3D alignment control program for complex spatial paths

Following the determination of the alignment geometry, the ring layout and detailed control program further resolve the complex assembly logic of components within the spatial sequence. To enhance structural integrity, shield tunnels typically employ a staggered jointing pattern, which necessitates the program to automatically calculate the optimal rotation angle for each segment ring based on predefined construction specifications. This module (as shown in Figure 6) utilizes logical decision nodes to dynamically control the circumferential positioning of the key segment (K-segment) and simultaneously derive the theoretical spatial orientation of circumferential and axial bolts. The program not only ensures the precise alignment of bolt holes between adjacent rings but also monitors the influence of alignment curvature on the staggered rotation step. Consequently, the automated optimization of the segment ring layout is achieved while strictly adhering to the constraints of assembly gaps and overlap requirements.

To instantiate the aforementioned macro-paths and meso-layout logic into model entities, the program finally implements a precise mapping of micro-positioning commands via a spatial coordinate resolution module for adaptive points. As shown in Figure 7, this module accounts for multivariate parameters, including the segment taper amount, vertical curve length, real-time rotation angle, and ring inclination. It utilizes spatial coordinate transformation matrices to derive the instantaneous pose of key control points for each segment in 3D space. Specifically, the program constructs a local Frenet Frame based on the tangent, normal, and binormal vectors of the centerline at the specific stationing. The resolved high-precision coordinates are then transmitted in real-time to the predefined parametric families. This "point-driven" mechanism enables each adaptive point to undergo precise spatial displacement, thereby completing the high-precision automated assembly of shield tunnel segments along complex alignments.

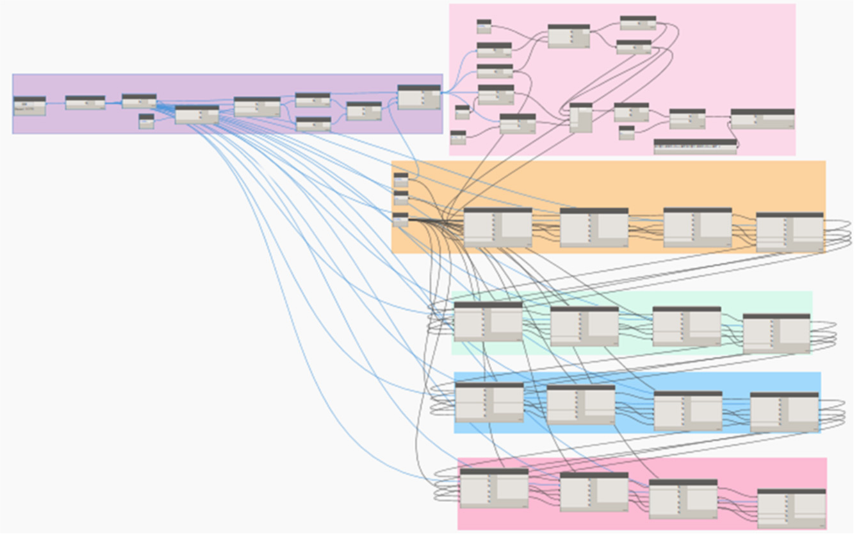


Fig 6. Workflow for automated segment ring layout and staggered assembly logic

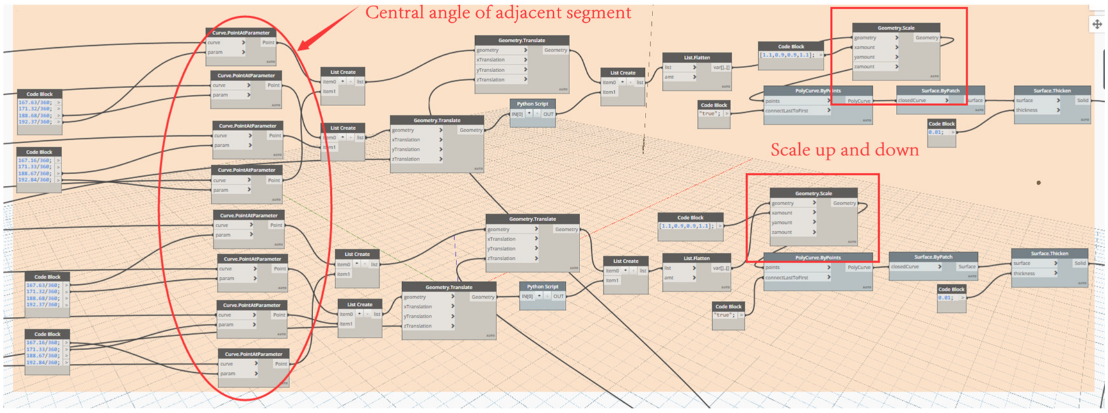


Fig 7. Spatial coordinate resolution and mapping mechanism for adaptive points

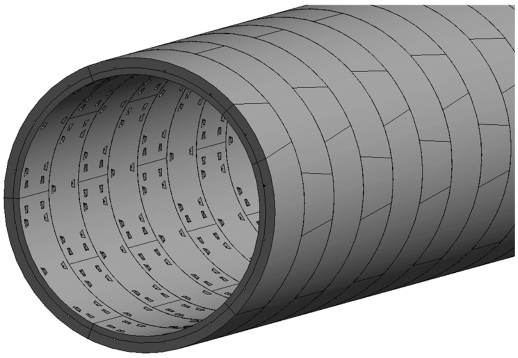


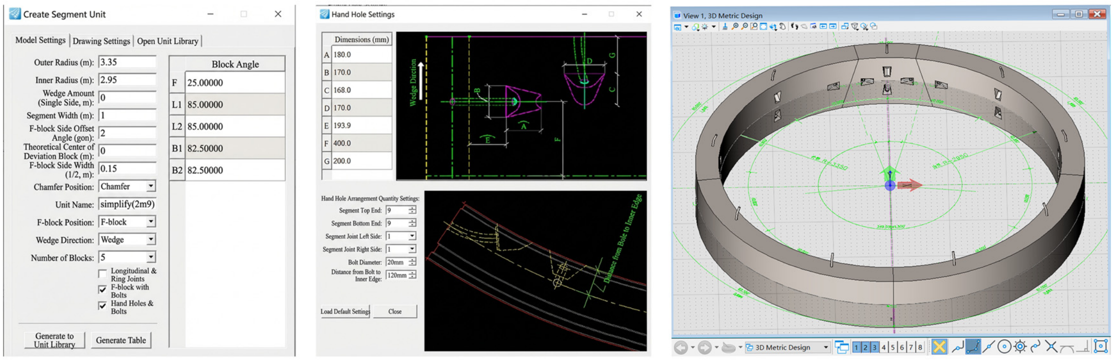
Fig 8. Parametric shield tunnel model generated through the Revit-Dynamo collaborative path

Through the deep synergy of the three aforementioned program modules, Dynamo serves not only as a logical bridge between external data and Revit geometric entities but also as a mechanism for the rigorous management of layout logic and coordinate resolution. This ensures high data integrity during design revisions and scheme comparison phases. Such a highly modular programming architecture allows designers to achieve a full-process closed-loop drive—ranging from alignment fitting to detailed component generation—exclusively through parameter adjustments. This approach significantly enhances both modeling efficiency and model quality. The parametric segment model constructed based on this workflow is illustrated in Figure 8.

4. Construction of Parametric Models Based on Existing Custom Development Tools

As a specialized BIM solution tailored for the railway and tunnel engineering sectors, OpenRail Designer For China provides robust underlying data support and geometric engine APIs for high-precision parametric design. This study utilizes a specialized plugin, "Shield Tunnel Segment Forward Design," developed by Beijing Huachuang Huixiang Technology Co., Ltd., based on the OpenRail platform. This approach enables the highly integrated and automated modeling of shield lining structures.

During the implementation of parametric modeling, the accurate definition of component geometric morphology is the prerequisite for achieving forward design. Through the interactive interface integrated into the plugin, designers can directly input core design parameters—including the outer and inner radii, ring width, segment quantity, taper amount, and subtended angles. These inputs establish the fundamental topological framework of the segments within the software's underlying logic. To meet the requirements of refined engineering management, this modeling path further extends to the meticulous control of detailed structures. Based on the established fundamental framework, the fine-grained configuration of parameters—such as bolt positioning, hand hole dimensions, and circumferential and longitudinal joint gaps—significantly enhances the information depth and geometric fidelity of the BIM model. The detailed process is illustrated in Figure 9.



Outer Radius (m): 3.35
Inner Radius (m): 2.95
Wedge Amount (Single Side, m): 0
Segment Width (m): 1
F-block Side Offset Angle (gon): 2
Theoretical Center of Deviation Block (m): 0
F-block Side Width (1/2, m): 0.15
Chamfer Position: Chamfer
Unit Name: simplify(2m9)
F-block Position: F-block
Wedge Direction: Wedge
Number of Blocks: 5
☐ Longitudinal & Ring Joints
☑ F-block with Bolts
☑ Hand Holes & Bolts

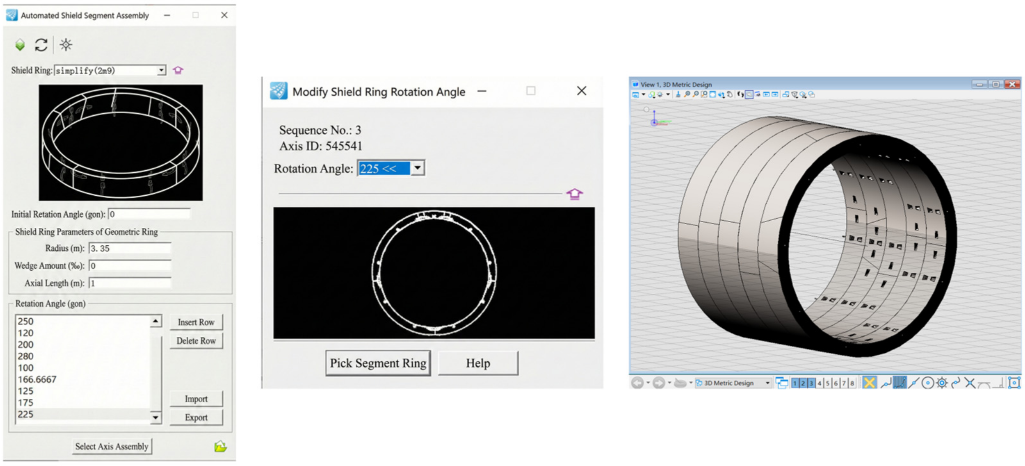
	Block Angle
F	25.00000
L1	85.00000
L2	85.00000
B1	82.50000
B2	82.50000

	Dimensions (mm)
A	180.0
B	170.0
C	168.0
D	170.0
E	193.9
F	400.0
G	200.0

Hand Hole Arrangement Quantity Settings:
Segment Top End: 9
Segment Bottom End: 9
Segment Joint Left Side: 1
Segment Joint Right Side: 1
Bolt Diameter: 20mm
Distance from Bolt to Inner Edge: 120mm

Fig 9. Parametric configuration and geometric definition of a single segment ring

The spatial positioning and automated generation logic of the model rely on the coupling of the design axis and assembly rules. After completing the parametric configuration of a single-segment ring by specifying the tunnel's assembly axis and presetting the staggered angles between rings, the system invokes underlying API interfaces to automatically perform coordinate resolution and solid Boolean operations. Once the centerline is exported and matched, the plugin drives the geometric engine to rapidly generate fully parametric shield segment assemblies in 3D space. This integrated development paradigm not only eliminates the cumulative errors inherent in manual modeling but also transforms complex underlying algorithms into intuitive workflows through the encapsulation of functional modules. Consequently, it significantly enhances the construction efficiency of BIM models for large-scale shield tunnels. The relevant modeling workflow and software interactive interface are illustrated in Figure 10.



Shield Ring: simplify(2m9)
Initial Retation Angle (gon): 0
Shield Ring Parameters of Geometric Ring
Radius (m): 3. 35
Wedge Amount (‰): 0
Axial Length (m): 1
Retation Angle (gon)
250
120
200
280
100
166.6667
125
175
225

Sequence No.: 3
Axis ID: 545541
Rotation Angle: 225 <<

Fig 10. Automated spatial assembly and positioning of large-scale segment assemblies

5. Conclusion

Aiming at the challenges of parametric modeling for shield tunnel segments, this study compares two distinct paths—visual programming-driven and underlying API-based integration—through case studies, yielding the following conclusions:

The modeling path based on visual programming (e.g., Revit + Dynamo) exhibits a high degree of flexibility and interactivity. By utilizing a "data-program-component" collaborative mode, this method achieves automated mapping from alignment parameters to adaptive families. Its primary advantages include a low development threshold and high logic transparency, enabling rapid responses to design changes. Consequently, it is particularly suitable for customized projects characterized by complex geometries and frequent scheme comparisons.

The modeling path based on underlying custom development (e.g., OpenRail plugins) demonstrates exceptional integration and standardization. By invoking underlying APIs to encapsulate core algorithms, this approach integrates cumbersome geometric calculations and layout logic into specialized plugins. When processing large-scale segment assemblies, this path shows superior computational stability and execution efficiency. It effectively unifies modeling standards and is ideal for mega-tunnel projects with high levels of standardization and massive datasets.

In summary, the two paths are complementary: visual programming focuses on agile development and the flexible application of logic, whereas API-based plugins lean toward deep functional accumulation and large-scale production. In practical engineering, technical routes should be selected scientifically based on project scale, complexity, and R&D capabilities to facilitate the digital transformation of shield tunnels from traditional "individual modeling" to efficient "forward design."

References

[1] WANG C, FRIEDMAN M, WU W, et al. Hydraulic influences on the long-term performance of tunnels: A review [J]. Transp Geotech, 2024, 48(16).

[2] SHA F, ZHANG M L, DONG Y L, et al. A review on the key factors influencing the stability of shield tunneling [J]. KSCE J Civ Eng, 2025, 29(11): 17.

[3] HE S Y, LAI J X, WANG L X, et al. A literature review on properties and applications of grouts for shield tunnel [J]. Constr Build Mater, 2020, 239(17).

[4] ZHANG J X, ZHANG Y X, REN X H, et al. Design and assembly workflow of universal segment of shield tunnel based on BIM [J]. KSCE J Civ Eng, 2025, 29(4): 13.

[5] CHENG Q, WANG X Y, SUN J B, et al. Research on BIM-Based Visualization, Simulation, and Early Warning System for Shield Tunnel Construction [J]. Buildings-Basel, 2025, 15(5): 18.

[6] BUI H G, NINIC J, KOCH C, et al. Integrated BIM-based modeling and simulation of segmental tunnel lining by means of isogeometric analysis [J]. Finite Elem Anal Des, 2024, 229(23).

[7] LUO H B, LI L H, CHEN K. Parametric modeling for detailed typesetting and deviation correction in shield tunneling construction [J]. Autom Constr, 2022, 134(12).

[8] ZHU X, BAO T F, YU H, et al. Utilizing Building Information Modeling and Visual Programming for Segment Design and Composition [J]. J Comput Civil Eng, 2020, 34(4): 13.

[9] HUANG H, RUAN B, WU X G, et al. Parameterized modeling and safety simulation of shield tunnel based on BIM-FEM automation framework [J]. Autom Constr, 2024, 162(22).

[10] HUANG Q F, LIU S P, LV Y G, et al. Modification of Segment Structure Calculation Theory and Development and Application of Integrated Software for a Shield Tunnel [J]. Appl Sci-Basel, 2022, 12 (12): 14.