

Document Version

Final published version

Licence

CC BY

Citation (APA)

Miculaş, C. V., Costa, R. J., da Silva, L. S., Simões, R., Craveiro, H., & Tankova, T. (2024). 3D macro-element for innovative plug-and-play joints. *Journal of Constructional Steel Research*, 214, Article 108436. <https://doi.org/10.1016/j.jcsr.2023.108436>

Important note

To cite this publication, please use the final published version (if applicable). Please check the document version above.

Copyright

In case the licence states “Dutch Copyright Act (Article 25fa)”, this publication was made available Green Open Access via the TU Delft Institutional Repository pursuant to Dutch Copyright Act (Article 25fa, the Taverne amendment). This provision does not affect copyright ownership. Unless copyright is transferred by contract or statute, it remains with the copyright holder.

Sharing and reuse

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights. We will remove access to the work immediately and investigate your claim.



3D macro-element for innovative plug-and-play joints

Cristian V. Miculaș^{a,*}, Ricardo J. Costa^a, Luis Simões da Silva^a, Rui Simões^a, Helder Craveiro^a, Trayana Tankova^b

^a Civil Engineering Department, ISE, University of Coimbra, 3030-788 Coimbra, Portugal

^b Department of Engineering Structures, Delft University of Technology, 2628 CD Delft, Netherlands

ARTICLE INFO

Keywords:

Innovative plug-and-play joints
Lightweight steel frames
Tubular columns
Macro-element
OpenSees

ABSTRACT

This paper presents the development, implementation, and validation of a macro-element suitable for the linear analysis of innovative 3D plug-and-play joints between tubular columns and lightweight steel truss-girders. The macro-element is based on the component method, accounts for the three-dimensional interaction between the tube faces, and its components have a clear physical meaning. Simplified procedures are developed for the closed-form computation of the stiffness matrix of the macro-element based on the geometric and mechanical properties of the nodal zone. This facilitates practical application in everyday design scenarios. Furthermore, the macro-element's architecture is implemented in the framework of OpenSees as a standalone beam-to-column joint finite element. Validation of the conceptual design is accomplished through parametric studies, comparing its performance with models generated in higher-order finite element commercial software, Abaqus. This research offers a valuable resource for the linear analysis and design of innovative 3D plug-and-play joint connections in structural engineering, enhancing efficiency and reliability in construction practices.

1. Introduction

In the framework of the Inno3DJoints project [1], an innovative three-dimensional (3D) plug-and-play moment resistant joint (P&PJ) was developed for modular construction systems consisting of tubular steel columns, cold-formed lightweight steel (CFS) truss-girders, and cross-laminated timber (CLT) slabs, as shown in Fig. 1. For clarity, a P&PJ system with only one truss-girder attached to the column is depicted; however, in the most general case, four truss-girders with different depths can be connected to the column. The main objective of the Inno3DJoints project was the development of a structural system that facilitates easy on-site (dis)assembly and transportation, enabling structural reuse. This innovative system was specifically designed for low to mid-rise buildings with a minimal carbon footprint in mind [2,3]. At its current stage, the technology readiness level (TRL) of the system stands at 4–5, signifying a relevant stage of development.

The Inno3DJoints plug-and-play joint system is composed of two parts that allow connecting the CFS truss-girder to the tubular column, named *socket* and *plug*, as shown in Fig. 2(a). The socket is made of two Z-shaped plates, cold bent, welded symmetrically to the column's face, and the plug consists of two parts, the *Y-fork* and the *T-plug*, which are welded together, as presented in Fig. 2(a). The Y-fork is adjusted to the width of the CFS truss-girder, and the T-plug has predrilled holes

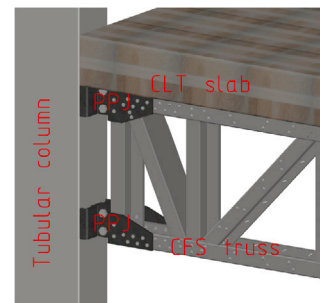


Fig. 1. The innovative 3D plug-and-play joint (P&PJ) system: overall view.

to facilitate the positioning inside the socket. These are aligned with the top and bottom chords of the CFS truss-girder, and thus, the main internal force applied on the joint by the CFS truss-girder is the axial force acting along the longitudinal axis of the chords.

During the fabrication process in the steel workshop, the T-plug and the Y-fork are welded to each other, the Y-fork is connected to the CFS truss-girder by a simple bolted lap joint, and the socket is welded to

* Corresponding author.

E-mail addresses: cristian.miculas@uc.pt (C.V. Miculaș), rjcosta@dec.uc.pt (R.J. Costa), luiss@dec.uc.pt (L.S. da Silva), rads@dec.uc.pt (R. Simões), heldercraveiro.eng@uc.pt (H. Craveiro), t.tankova@tudelft.nl (T. Tankova).

<https://doi.org/10.1016/j.jcsr.2023.108436>

Received 1 November 2023; Received in revised form 15 December 2023; Accepted 22 December 2023

Available online 2 January 2024

0143-974X/© 2023 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

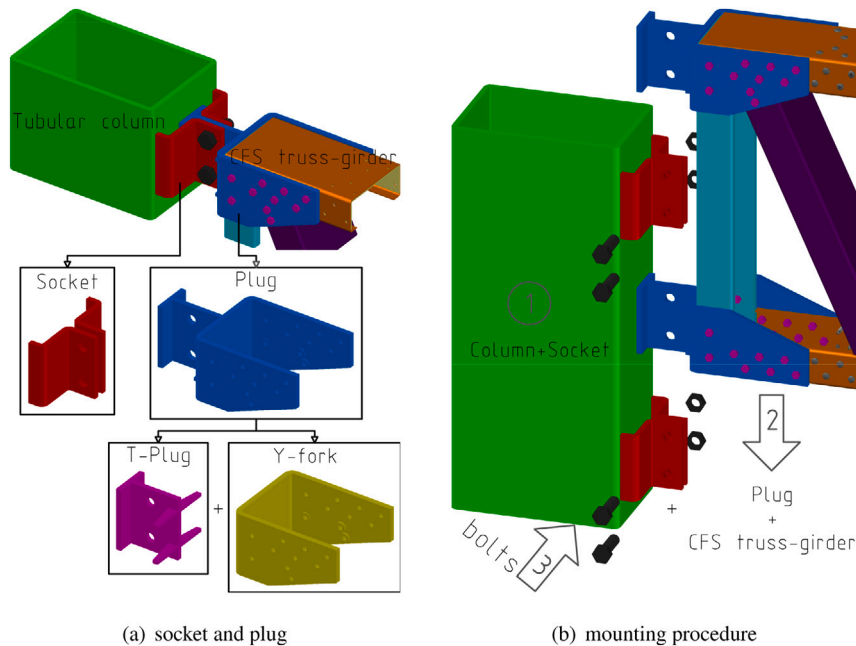


Fig. 2. The innovative 3D plug-and-play joint system: socket and plug.

the column, as shown in Fig. 2(a). This joint configuration allows for an easy assembly, as illustrated in Fig. 2(b), following these sequential steps [3,4]:

1. the columns with the sockets previously welded to them are placed in position;
2. the T-plug (with the CFS truss-girders attached to it) is inserted vertically, i.e. descending movement, in the existing gap between the two pieces of the socket;
3. the T-plug and socket are bolted together;
4. the CLT slab is positioned and fastened to the CFS truss-girders.

Alternatively, the CFS truss-girders could be connected to the CLT slab after the first step. Then, this entire system is inserted vertically in position and bolted to the column — this reduces the construction site hazards and increases the offsite prefabrication but could lead to higher transportation costs.

In this paper, a mechanical model based on the component method for the P&PJ system, compatible with Eurocode 3, Part 1-8 [5], is presented and validated. Moreover, the model is used to derive a novel macro-element that incorporates the 3D behaviour of tubular columns. Subsequently, this macro-element is utilised to create a dedicated finite element for beam-to-column P&PJ configurations. This finite element is designed to facilitate the analysis and design of P&PJ systems within the linear regime, specifically tailored for everyday design purposes. Finally, this macro-element is implemented in OpenSees [6,7] as a standalone beam-to-column joint finite element. For enhanced readability of the work presented in this paper, a flowchart illustrating the relationship between the analytical and finite element models and their corresponding sections in the article is provided in Appendix D

2. State-of-the-art

2.1. Component method

The component method [8] is a simple, intuitive and versatile procedure that evaluates the mechanical properties of the joints by considering the diversity of the structural systems that result from various load configurations, including static, dynamic and fire actions. The originality of the component method is that it considers any joint

as a set of individual basic components that represent the effect of a stress resultant in a physical portion of the joint.

The component method can be applied with different degrees of refinement, ranging from dimensionless OD lumped elements *rotational springs* or *linear springs* to 3D complex macro-elements [9–13] and different structural materials [14–16]. Nowadays, it is widely accepted by the design community due to its implementation in EN 1993-1-8 [1].

The component method relies on three steps [17]:

1. identification of the *basic components* or simply *components*, i.e. the pairs part of the joint and internal forces that are relevant to one, or more, mechanical properties of the joint, namely strength, stiffness or ductility;
2. characterisation of the intrinsic behaviour of each component, e.g., the force deformation of each part of the joint;
3. assembly of the components, i.e. the definition of the spatial arrangement and connectivity of the components.

The output of the component method is a mechanical model, i.e. a macro-element model, that simulates the behaviour of the joint under the stress fields for which it was developed. However, for ease of implementation, some simplifications are usually taken. For example, instead of a macro-element model, a lumped model is considered where the behaviour of the joint is accounted for through rotational OD elements positioned at the beams' ends.

Therefore, to achieve a higher degree of consistency between the classical design procedure for traditional beam-to-column steel joints and the P&PJ system, a macro-element that follows the component method is developed for the design and analysis of the P&PJ system proposed in the Inno3DJoints project [1].

2.2. Macro-element modelling approach

The macro-element modelling technique of beam-to-column joints has gained significant attention in recent years due to its ability to analyse the joint independently from the connecting members in a structure. This technique is commonly used in connections between steel members with I/H cross-sections. In these connections, the behaviour of the joint is not represented by rotational OD elements at the ends of the beams but rather is explicitly modelled using a system of

multiple OD elements and rigid links, each representing a specific component with its own model. As a result, the multiple joints, connections, and web panel are treated together in a single joint macro-element.

Several authors have applied this technique. For example, Bayo et al. [18–22] devised multiple 2D macro-elements for internal and external semi-rigid joints between I/H elements using finite-sized deformable beam elements for global analysis of frames. Similarly, Costa et al. [23] formulated two-dimensional macro-element models for joints with equal and unequal beam depths between I/H elements using lumped OD linear elements combined with rigid links, which Simões da Silva et al. [13] expanded to 3D joints.

Harada and Simões da Silva have conducted extensive research on the modelling of I/H beam-to-tubular column joints without diaphragm plates [24–26]. They introduced a new joint component called the cross-section component, which represents the local deformation of the rectangular hollow section (RHS) in the beam-to-column joint region. The behaviour of the joint in this region is complex because the out-of-plane behaviour of the column plates is interdependent. Additionally, they developed and validated a macro-element that includes the new cross-section component for 3D joints. The macro-element was able to accurately reproduce the interaction between the four column plates in I/H beam-to-RHS column joints under various loading configurations, particularly when the joint behaviour was nonlinear.

2.3. Existing beam-to-column joint finite elements

Although the macro-element modelling technique is frequently used by researchers, their conceptual macro-elements are rarely implemented as joint finite elements in finite element software. This limitation hinders their utilisation for global analysis of structures and restricts their accessibility to a broader community of users.

For example, the joint element library of the OpenSees [27] has only four joint finite elements:

1. a beam-to-column joint model for simulating the earthquake response of reinforced concrete frames in 2D nonlinear analysis [28,29];
2. a beam-to-column joint for simulation and damage models for performance assessment of reinforced concrete joints in 2D and 3D, under cyclic loading, with the capability of integrating both geometric and material nonlinearities [30];
3. a 2D planar panel zone element [31], whose analytical formulation is also implemented in HybridFEM software and used for dynamic time history analysis of 2D inelastic framed structures and real-time hybrid simulation [32];
4. a 2D elastic tubular joint that incorporates the effects of the joint's flexibility to produce a more realistic response for offshore structures [33].

However, the joint element library of OpenSees can be considered extensive when compared to other commercial finite element software [34–37], as it includes several finite elements for modelling beam-to-column nodes. In conclusion, there is a need to create and utilise joint finite elements, particularly for steel and steel-concrete composite beam-to-column joints and for 3D applications.

3. Macro-modelling of P&PJ systems with one CFS truss-girder

The application of the component method to the Inno3DJoints project [38] resulted in a mechanical model that enables smooth analysis and design of structures built with the P&PJ system. The relevant components for the simplest P&PJ configuration, in which only one truss-girder is connected to the column, are shown in Fig. 3. Additionally, Fig. 3 shows the component model for a joint whose mechanical properties can be found in [38] – the components of this model can

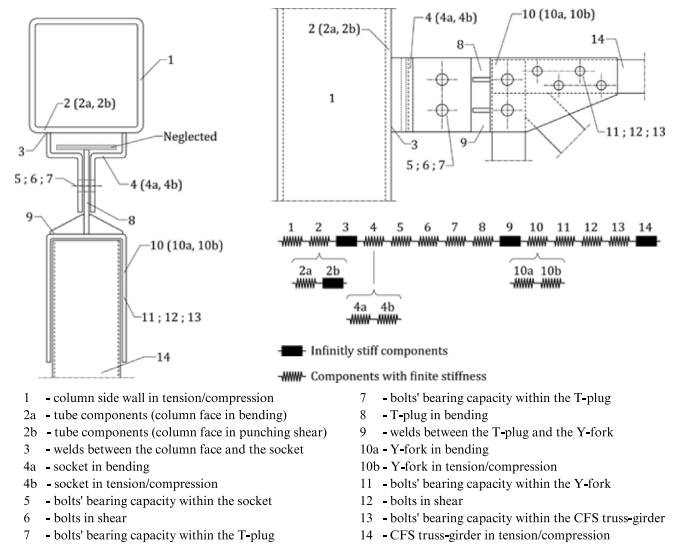


Fig. 3. Component model for a P&PJ with one truss-girder: top view (left), side view (top right) and simplified representation of the component model (bottom left). Source: Adapted from [39].

easily be lumped into a single OD element, i.e. a linear spring, as follows:

$$\frac{1}{k_{\text{lumped}}} = \sum \frac{1}{k_i} \quad (1)$$

where k_i is the stiffness of each component [39]. The strength of the lumped OD element, f_{lumped} , is the minimum strength of all the components, thus:

$$f_{\text{lumped}} = \min(f_i) \quad (2)$$

where f_i is the strength of each component [39].

In P&PJ configurations with multiple CFS truss-girders connected to the column, the determination of the stiffness and strength of the tube components relies on the simultaneous application of forces on all tube faces. This necessity arises from the achievement of a 3D interaction between the column faces, which is demonstrated by both experimental findings and numerical investigations [25,26,40].

Hereinafter, the relevant components of the P&PJ system are classified into two categories: *connection components* and *tube components*. The connection components describe the mechanical behaviour of the column side wall, welds, socket, bolts, plug, and CFS truss-girder's ends, while the tube component represents the mechanical behaviour of the tube faces and their interaction.

4. Macro-element development for 3D interaction

4.1. Column macro-element concept

In terms of individual components, the difference between configurations with one or more truss-girders connected to the column is that, in the latter case, the tube components are treated separately and not included in the arrangement shown in Fig. 3.

The 3D interaction between the column faces is accounted for using a macro-element approach. To consider both bending and shear deformation of the column segment between the two joints and/or to analyse truss-girders with different depths, it was found more suitable to model each joint independently with separate macro-elements, as shown in Fig. 4(a). Finally, the behaviour of the tube components is described by the mechanical model shown in Fig. 4(b), but without the inclusion of the connection components, and referred to as the *column macro-element* (CME).

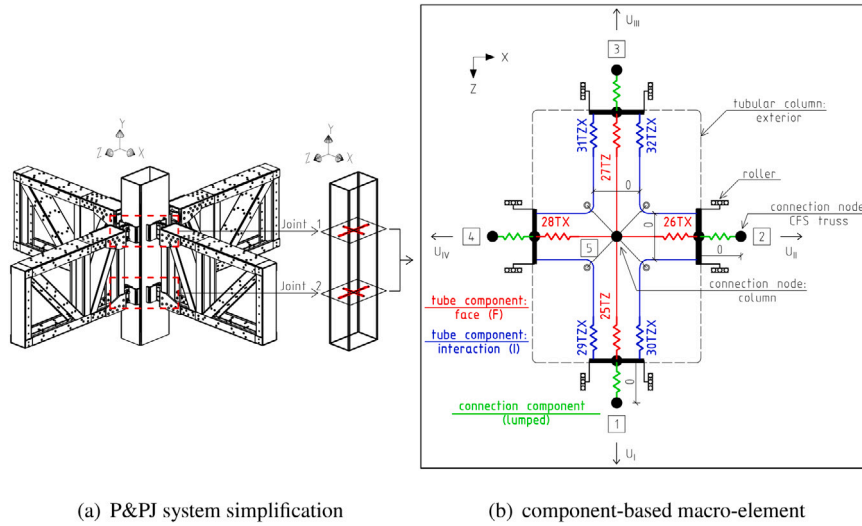


Fig. 4. Macro-element modelling approach of the P&PJ system.

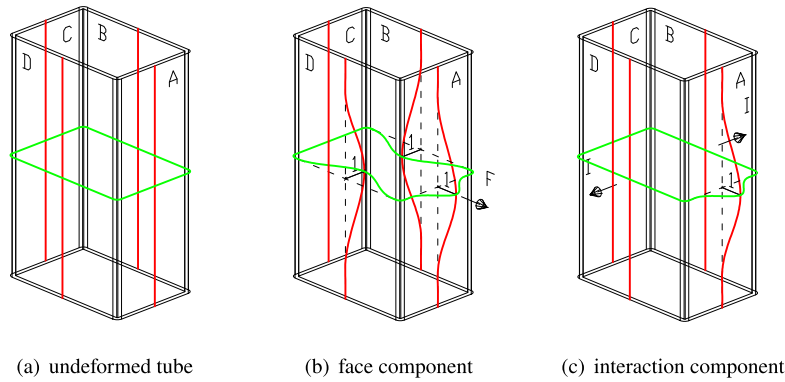


Fig. 5. Definition of the stiffness of the tube components.

The CME has two types of components: the *face component* (F) and the *interaction component* (I). The physical meaning of these components can be understood through the definition of their stiffness for a generic column face, i.e. face A, shown in Fig. 5:

- face component’s stiffness (k_F) – the force that produces a unitary outward out-of-plane displacement on face A, when simultaneously unitary inward out-of-plane displacements are imposed on the adjacent faces, i.e. on faces B and D in Fig. 5(b);
- interaction component’s stiffness (k_I) – the force that keeps null the out-of-plane displacements on two parallel faces, e.g. faces B and D in Fig. 5(c), while a unitary out-of-plane displacement is imposed on the analysed face of the tube, i.e. on face A of Fig. 5(c).

Based on this definition, it can be seen that a square cross-section requires only one type of face component, k_F , because all tube faces have the same length. However, the rectangular cross-section has two different face lengths, requiring the use of two types of face components: $k_{F,L1}$ and $k_{F,L2}$. On the other hand, only one type of interaction component, k_I , is needed for both cross-sections.

4.2. Column macro-element validation

4.2.1. General procedure

To validate the concept of the CME, the following steps were taken:

1. computation of the stiffness of the tube components using an higher-order finite element model; in this case, Abaqus [34] was

used and the resulting model is referred to as the refined finite element model (RFEM);

2. assembling of the stiffness matrix for the CME model;
3. selection of a load pattern (LP);
4. computation of the out-of-plane displacements of the column faces for the selected LP using the RFEM;
5. computation of the out-of-plane displacements of the column faces for the selected LP using the CME model;
6. comparison of the results between models (RFE vs. CME) – ideally, they should be identical or within an acceptable range of error.

Following the notations of the column faces from Fig. 5, the stiffness of the face components for faces A and C, and B and D becomes k_F^{AC} and k_F^{BD} , respectively. The stiffness of the interaction component is denoted as k_I . For the set of coordinates represented in Fig. 4(b), the out-of-plane displacements, U_{CME} , of the column faces are:

$$U_{CME} = [U_I \quad U_{II} \quad U_{III} \quad U_{IV}]^T \quad (3)$$

To remove the rigid body movements of the CME, it was assumed that node 5 (Fig. 4(b)) is fixed; thus, the stiffness matrix for the beam-to-column joint in this specific case is:

$$K_{CME} = \begin{bmatrix} k_F^{AC} + 2 \times k_I & k_I & 0 & k_I \\ k_I & k_F^{BD} + 2 \times k_I & k_I & 0 \\ 0 & k_I & k_F^{AC} + 2 \times k_I & k_I \\ k_I & 0 & k_I & k_F^{BD} + 2 \times k_I \end{bmatrix} \quad (4)$$

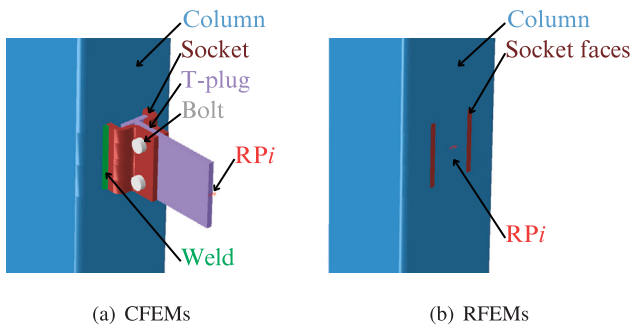


Fig. 6. Schematic representation of the Abaqus models.

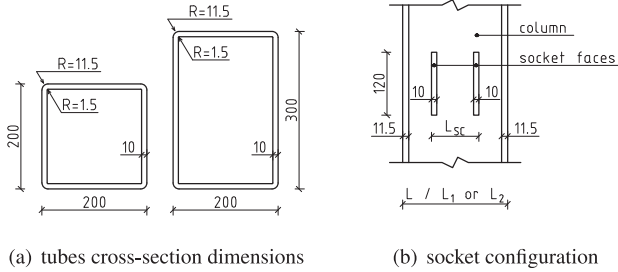


Fig. 7. Geometric properties.

For a random LP, the applied forces, \mathbf{P}_{CME} , are:

$$\mathbf{P}_{\text{CME}} = [P_I \quad P_{II} \quad P_{III} \quad P_{IV}]^T \quad (5)$$

Therefore, the out-of-plane displacements of the column faces, \mathbf{U}_{CME} , become:

$$\mathbf{U}_{\text{CME}} = \mathbf{K}_{\text{CME}}^{-1} \cdot \mathbf{P}_{\text{CME}} \quad (6)$$

4.2.2. Numerical models

The conceptual validation of the CME for the P&PJ system was conducted using computational models developed within the Abaqus software framework [34].

While conventional practice typically involves the creation of complete finite element models (CFEMs) encompassing a comprehensive representation of all system components, including the column, socket, T-plug, bolts, and welds, a deliberate departure from this standard approach was undertaken in favour of constructing the most streamlined and refined finite element models (RFEMs) feasible.

This modified approach resulted in the construction of models representing the column and the two specific regions corresponding to the contact interface between the socket and the column, referred to as *socket faces*, as represented in Fig. 6(b).

The decision to adopt this approach was driven by the substantial computational time demands associated with CFEMs, which were necessitated by the comprehensive analysis outlined in Section 6. Further details explaining the rationale behind this simplification will be provided later in this section.

It is important to note that the modelling effort concentrated exclusively on a single joint, specifically the top chord or 'Joint 1', as depicted in Fig. 4(a).

Refined Finite Element Models (RFEMs) The specimens chosen for concept validation featured a square hollow section (SHS) column with dimensions as shown in Fig. 7(a) and a height of 3000 mm. The column material followed a linear-elastic constitutive relation defined by Young's modulus, $E = 210$ GPa, and Poisson's ratio, $\nu = 0.3$ [41].

The column part was built with solid brick elements employing reduced integration (C3D8R), whereas the socket faces are represented by

shell discrete rigid elements (R3D4). In compliance with the software requirements, a reference point (RP) is defined at the centre of the socket faces, as depicted in Fig. 6(b) with i denoting the identifier for the column face. It is noteworthy that the motion of the rigid body is solely governed by this RP, thereby ensuring uniform translation and rotational behaviour for all points situated on the socket faces.

The solid and shell elements are linked using a surface-to-surface tie kinematic constraint, effectively preventing any relative motion between these two surfaces.

The column part exhibits rounded corners, with a fully-fixed boundary condition applied to mid-rounding nodes along its height to prevent rigid body motions. Additionally, in-plane rotation of the RP is restrained.

The forces and displacements simulating the demands coming from the socket are assigned to the RPs.

In the proximity of the joint, at a distance equal to the socket's height, the column's mesh size is maintained at approximately 5 mm. Beyond this region, a single, biased mesh strategy is applied to the column's edges, resulting in a gradual transition from 5 mm at the joint region to 50 mm at the column's ends. The mesh assignment is shown in Fig. 8. Additionally, to prevent shear locking, four elements along the thickness of the column were considered.

The main characteristics of the RFEMs outlined above result from a series of parametric studies involving mesh, geometry, and boundary conditions, detailed in the subsequent section. These studies aimed to achieve mesh-independent results and ensure the local behaviour of the tube within the joint region remained unaffected by boundary conditions, mesh size, and column height variations. These calculations were performed for LP '01' from Fig. C.1 – and a socket configuration with $L_{SC} = 20$ mm (Fig. 7(b)). It is noteworthy that although the parametric studies focused on SHS, their conclusions are applicable to RHS as well.

The generation of RFEMs using Abaqus, as exemplified earlier and employed in this research, can be easily accomplished using the 'abqTCphd' Abaqus RSG plug-in [42] developed in Python [43]. This plug-in comprises five modules, namely geometry, material, mesh, load, analysis, and job, providing a comprehensive set of parameterisation capabilities. Consequently, users are empowered to fully customise and adjust various aspects of the simulation process. Furthermore, comprehensive video tutorials are available on YouTube, covering the processes of downloading and installing the plug-in [44], as well as generating models using it [45].

The influence of mesh size on the results was investigated by testing three mesh sizes (10.0 mm, 5.0 mm and 2.5 mm) for the column.

The out-of-plane displacements of the loaded column face along the symmetry longitudinal section ('H1') and a symmetry transversal cross-section ('W0') of the column for the considered mesh sizes are shown in Figs. 9(a) and 9(b), respectively. Notably, both 'H1' and 'W0' exhibit identical, mesh-independent results for mesh sizes of 5.0 mm ('m 5.0') and 2.5 mm ('m 2.5'). Consequently, a mesh size of 5.0 mm is determined as the upper limit for the uniform mesh configuration.

Nevertheless, an exploration into variations in mesh size along the column's height was undertaken by implementing localised seed and bias parameters, aiming to reduce the total number of finite elements and computational demands. Consequently, the mesh size was set to 5.0 mm in the joint region and progressively increased to 50.0 mm towards the column's extremities ('m 5.0 w/ bias'). As it can be seen in Fig. 9 this model with biases and seeds ('m 5.0 w/ bias') provides the same results when compared to the uniform mesh model ('m 5.0'). Therefore, 'm 5.0 w/ bias' is established as the final adopted mesh configuration for the column in this study, as further detailed in Fig. 8.

A preliminary investigation conducted using simplified cross-section models indicated that the deformation of the column's cross-sections, due to out-of-plane forces applied to the column faces, results in negligible displacements at the external corners of the tubes. Consequently, a method to distinguish between the displacements caused by

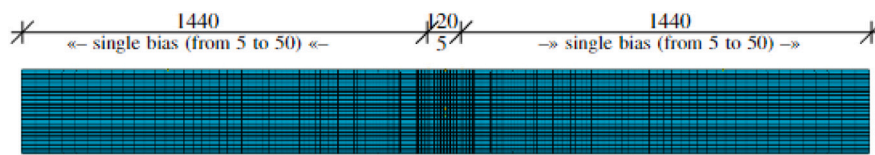


Fig. 8. Tubular columns: mesh assignment.

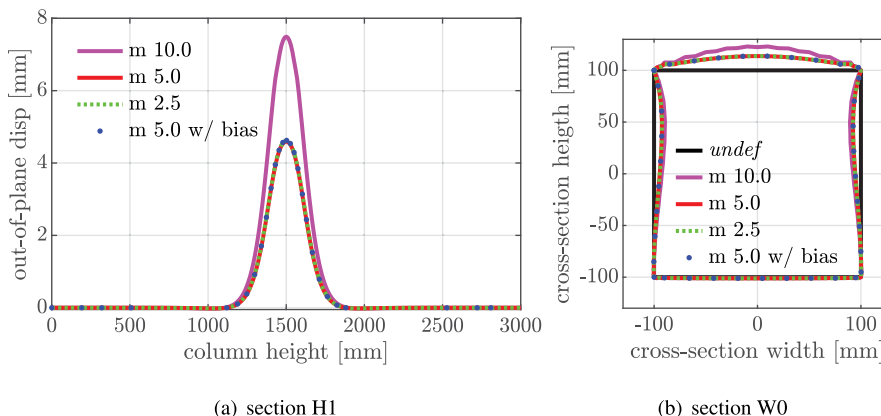


Fig. 9. Column out-of-plane deformation under different uniform mesh sizes.

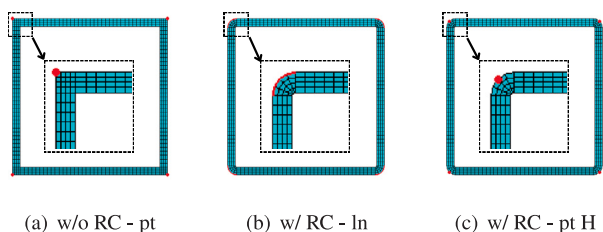


Fig. 10. BC and rounded corners analysis: cross-sectional view of the column.

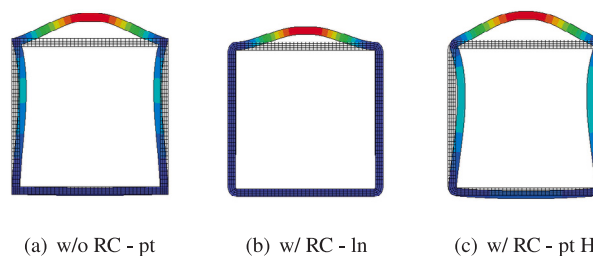


Fig. 11. BC and rounded corners analysis: deformed shape (U magnitude) in section W0.

cross-sectional deformation and the rigid body displacements of these cross-sections in the FE models is to apply displacement constraints to the nodes located at the external corners of the columns. However, it is imperative to acknowledge that the considered tubes have rounded corners, a characteristic inherent to their cold-bending manufacturing process. The objective of this investigation was to ascertain the magnitude of deformation in tube cross-sections induced by out-of-plane forces on column faces and formulate approaches for confining these displacements within the FE models, considering the rounded corners of the tubes.

Three models were developed, one with straight corners ('w/o RC - pt') and the other two with rounded corners. The presence of the rounded corners introduced ambiguity in the application of boundary conditions. To prevent rigid body movement of the cross-section and maintain other deformation modes, fixed boundary conditions were tested in two distinct regions: (i) along the edges of the rounded corners ('w/ RC - ln') and (ii) at the mid-rounding nodes ('w/ RC - pt H'). These three distinct boundary condition scenarios were applied uniformly across the entire height of the column and are visually represented at the cross-sectional level in Fig. 10.

The overall deformation of the cross-sections is shown in Fig. 11 for all three models. Fig. 11(b) demonstrates that by constraining the entire rounded corner's region, the interaction between the column faces is prevented, causing the column faces to behave like four independently loaded plates rather than a tubular cross-section. As a result, RFEMs with this type of boundary condition were not considered further in this work.

The comparison between Figs. 11(a) and 11(c) shows that the inclusion of rounded corners has a notable impact on the displacements in the loaded column face and, thus, should be considered when conducting further analysis. Specifically, it was found that there was a 7.96% increase in displacement on the loaded face, which also had a ripple effect on the adjacent faces. However, the RFEM with straight corners (Fig. 11(a)) presents excessive local distortion in the proximity of the corners. This phenomenon results from using finite elements with a reduced integration scheme combined with the type and position of the applied boundary condition. As suggested by the software's manual [34], a potential solution to this problem is the use of finite elements with hybrid formulation, e.g., C3D8H. Nevertheless, the model with straight corners is not used beyond this paper section. Therefore, the geometry and BC application of the model 'w/ RC - pt H' (Fig. 10(c)) represents the adopted final configuration.

Furthermore, to evaluate the effectiveness of the 'w/ RC - pt H' model in preventing rigid body movement while preserving the cross-sectional deformation modes, a new model, 'w/ RC - pt E', was created. In this new model, the displacement restrictions are applied solely to the nodes at the column's ends. Fig. 12 illustrates the 3D deformed shapes of the full models, while Fig. 13 displays the deformations of their cross-sections in the 'W0' section in 2D.

To compare the deformation shapes of the cross-sections in 'w/ RC - pt H' and 'w/ RC - pt E' models, the rigid body movement component of the cross-section was subtracted from the latter model. The deformed

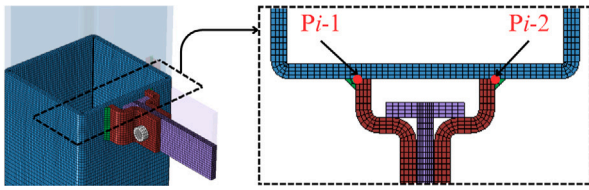


Fig. 15. Points locations on the column for out-of-plane displacement measurement.

Table 1

Results validation: RFEMs vs. CFEMs with 1 beam [mm].

t_c	RFEMs RP1	CFEMs P1	ϵ_1
10	-4.27	-4.55	6.20
8	-8.79	-9.30	5.46
6	-19.28	-20.37	5.32
4	-44.48	-48.53	8.35

Table 2

Results validation: RFEMs vs. CFEMs with 2 beams [mm].

t_c	RFEMs		CFEMs		ϵ_1	ϵ_2
	RP1	RP2	P1	P2		
10	-2.84	2.84	-3.01	3.01	5.58	5.58
8	-5.25	5.25	-5.58	5.58	6.07	6.07
6	-11.62	11.62	-12.37	12.37	6.04	6.04
4	-33.37	33.37	-34.75	34.75	3.95	3.97

Table 3

Tube components stiffness [kN/m].

SC	L_{SC} [mm]	k_P^{AC}	k_P^{BD}	k_1
01	20	62.33	10.87	12.58
02	40	78.74	9.97	15.96
03	90	187.32	0.75	34.27

The differences between RFEMs and CFEMs results were quantified using ϵ_i as defined in Eq. (7).

$$\epsilon_i = \frac{P_i - RP_i}{P_i} \times 100 \quad (7)$$

Analysis of the results in Tables 1 and 2 revealed that for configurations with one beam, the discrepancy between RFEMs and CFEMs fell within the range of 5.32% to 8.35%, and for configurations with two beams, it varied from 3.95% to 6.07%. These error ranges are considered acceptable, supporting the use of RFEMs as an alternative to CFEMs.

4.2.3. Validation of the CME using RFEMs

The validation employed three simplified RFEMs, featuring a rectangular hollow section (RHS) as depicted in Fig. 7(a). These RFEMs were distinguished by distinct socket configurations (SCs), primarily defined by the variable distance between the socket faces, L_{SC} (Fig. 7(b)). Of significance is the fact that all column faces shared the same SC. The SC naming, the distance between the sockets and the stiffness of the column components obtained from RFEMs using the procedure from Section 4.1 are presented in Table 3.

To validate the concept of the CME, three LPs were used: '02 CV', '03 V01', and '04 H', from Fig. C.2. These LPs respectively represent the effects in the top chord joint of gravity loads on a corner joint, the effects of gravity loads on an external joint, and the effects of horizontal loads on an internal joint. Each LP was assigned a dummy value of $P = 1.000$ kN.

The out-of-plane displacements of each tube face are presented in Table 4 for both RFEMs and CME across all SC and LP cases. The table also displays relative errors, ϵ_j (where j represents the column face designation, i.e. I, II, III, or IV), as defined by Eq. (8). These relative

errors are consistently low, primarily originating from deformation modes not considered by the CME model. This suggests that the CME can effectively account for the 3D interaction between the column faces.

$$\epsilon_j = \frac{U_{RFE} - U_{CME}}{U_{RFE}} \times 100 \quad (8)$$

4.3. Extended macro-element

The usability of the macro-element (Fig. 4(b)) was extended and integrated into a *beam-to-column joint finite element* that considers both tube and connection components. The new joint finite element was developed to allow modelling other types of loads and deformations not accounted for explicitly, e.g., the transversal deformation and rotation in each connection. Accordingly, in addition to the macro-element shown in Fig. 4(b), the beam-to-column joint finite element also includes 3D springs with six DOFs, corresponding to three linear springs and three rotational springs, enabling the modelling of any desired behaviour for the connection. The beam-to-column joint finite element is depicted in Fig. 16.

5. Beam-to-column joint finite element formulation

5.1. Overview

The beam-to-column joint finite element has the following main characteristics:

- five external nodes (1 to 5) with six DOFs each (DOF 1 to DOF 30), which are compatible with traditional 1D frame elements used to model columns and truss-girders where:
 - nodes 1 to 4 connect the chords of the CFS truss-girders to the column;
 - node 5 connects the column;
- four internal nodes with one DOF each (iDOF 1 to iDOF 4), which are required to create a kinematically determined element;
- thirty-two components represented by OD elements, as described in Table 5, where XYZ refers to the coordinate system used for the implementation of the element, as seen in Fig. 16.

5.2. Mathematical formulation

For a first-order analysis and a linear elastic behaviour assigned to the joint finite element's components (Fig. 16), the equilibrium relation between nodal forces and internal forces in the components can be expressed as:

$$\mathbf{f} = \mathbf{B} \cdot \mathbf{P} \quad (9)$$

where \mathbf{f} (32×1), \mathbf{B} (32×34), and \mathbf{P} (34×1) are the vector of internal forces in components, equilibrium matrix, and vector of nodal forces, respectively. Accordingly,

$$\mathbf{f} = [f_1 \quad \dots \quad f_{32}]^T \quad (10)$$

where f_i is the internal force in component i , and

$$\mathbf{P} = [P_{\text{ext},1} \quad \dots \quad P_{\text{ext},30} \quad P_{\text{int},1} \quad \dots \quad P_{\text{int},4}]^T \quad (11)$$

where $P_{\text{ext},i}$ and $P_{\text{int},j}$ are the nodal force in external DOF i and nodal force in internal DOF j , respectively.

The linearised compatibility relation between the deformations in the components and the nodal displacements can be expressed as:

$$\mathbf{\Delta} = \mathbf{A} \cdot \mathbf{U} \quad (12)$$

Table 4
Results validation: RFE vs. CME [mm].

LP	SC	RFE (Abaqus)				CME (column macro-element)				ϵ_j			
		U_I	U_{II}	U_{III}	U_{IV}	U_I	U_{II}	U_{III}	U_{IV}	ϵ_I	ϵ_{II}	ϵ_{III}	ϵ_{IV}
01	01	3.661	26.152	7.914	1.494	3.557	26.241	7.871	1.506	2.84	-0.34	0.55	-0.79
	02	3.216	22.743	5.935	1.047	3.140	22.826	5.898	1.051	2.38	-0.36	0.63	-0.31
	03	1.959	14.324	2.015	0.088	1.925	14.404	1.983	0.028	1.72	-0.56	1.61	67.52
02	01	-9.204	21.201	9.204	6.446	-9.308	21.247	9.308	6.500	-1.13	-0.21	-1.13	-0.85
	02	-7.091	18.370	7.091	5.420	-7.170	18.413	7.170	5.463	-1.11	-0.24	-1.11	-0.79
	03	-2.646	11.734	2.646	2.677	-2.687	11.774	2.687	2.658	-1.56	-0.34	-1.56	0.71
03	01	11.576	27.647	11.576	27.647	11.429	27.747	11.429	27.747	1.27	-0.36	1.27	-0.36
	02	9.151	23.790	9.151	23.790	9.037	23.876	9.037	23.876	1.24	-0.36	1.24	-0.36
	03	3.975	14.412	3.975	14.412	3.908	14.432	3.908	14.432	1.66	-0.14	1.66	-0.14

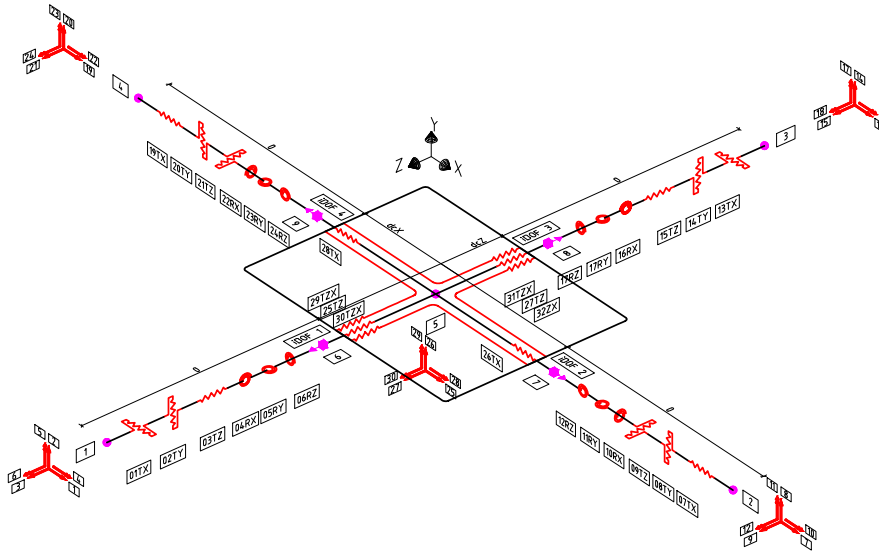


Fig. 16. Extended macro-element: beam-to-column joint finite element.

Table 5
Components deformations direction in the local coordinate system of the element.

Components	Deformation mode
1, 7, 13, 19, 26, 28	Linear deformation in X direction
2, 8, 14, 20	Linear deformation in Y direction
3, 9, 15, 21, 25, 27	Linear deformation in Z direction
29, 30, 31, 32	Linear deformation in ZX plane
4, 10, 16, 22	Angular deformation after X axis
5, 11, 17, 23	Angular deformation after Y axis
6, 12, 18, 24	Angular deformation after Z axis

where $\Delta(32 \times 1)$, $A(32 \times 34)$, and $U(34 \times 1)$ are the vector of deformation in components, compatibility matrix, and vector of nodal displacements, respectively. Therefore,

$$\Delta = [A_1 \quad \dots \quad A_{32}]^T \quad (13)$$

$$U = [U_{ext,1} \quad \dots \quad U_{ext,30} \quad U_{int,1} \quad \dots \quad U_{int,4}]^T \quad (14)$$

where A_l is the deformation in component l , and $U_{ext,i}$ and $U_{int,j}$ are the generalised displacement in external DOF i , and the displacement in internal DOF j , respectively.

Each row i of the compatibility matrix A shows the deformation in the components when a unitary generalised displacement is assigned to DOF i , while the other DOFs are kept null, e.g., element a_{ij} is the deformation in component i when a unitary generalised displacement is assigned to DOF j , while the other DOFs are kept null.

The following sign convention is used to compute the deformations in components:

- axial deformation components — positive for elongation and negative for shortening;
- shear deformation components — positive according to the sign convention from the strength of materials, assuming that the OD element is oriented according to the local reference axis parallel to it;
- bending deformation components — positive according to the sign convention from the strength of materials assuming that the OD element is oriented according to the local reference axis parallel to it;
- torsion deformation components — positive if, according to the right-hand rule, the rotation vector points outward from the component and negative if it points inward.

In order to make the reading and writing process of this document easier, only the non-zero elements of the compatibility matrix, A , of the proposed joint finite element (Fig. 16) are included in Table 6.

Finally, the constitutive relation of the components can be written as:

$$f = k_j \cdot \Delta \quad (15)$$

where $k_j(32 \times 32)$ is a diagonal matrix with $k_j(i, i)$ being the stiffness of component k .

In the scope of small deformations, the dual relation between compatibility and equilibrium leads to the following:

$$P = A^T \cdot f \quad (16)$$

By introducing Eqs. (12) and (15) into Eq. (16), the vector of external forces becomes:

$$P = A^T \cdot k_j \cdot A \cdot U = K_j \cdot U \quad (17)$$

Table 6
The non-zero elements of the compatibility matrix **A**.

Value	Elements of compatibility matrix A
1	$a_{1,1}; a_{2,2}; a_{3,3}; a_{5,5}; a_{6,6}; a_{7,7}; a_{9,9}; a_{10,10}; a_{11,11}; a_{12,12}; a_{16,16}; a_{20,20}; a_{8,26}; a_{13,25}; a_{14,26}; a_{16,28}; a_{17,29}; a_{18,30}; a_{21,27}; a_{22,28}; a_{23,29}; a_{24,30}; a_{15,33}; a_{19,34}; a_{25,31}; a_{26,32}; a_{27,27}; a_{28,25}; a_{29,25}; a_{31,25}; a_{31,27}; a_{32,27}; a_{29,31}; a_{32,32}$
-1	$a_{4,4}; a_{8,8}; a_{13,13}; a_{14,14}; a_{15,15}; a_{17,17}; a_{18,18}; a_{19,19}; a_{21,21}; a_{22,22}; a_{23,23}; a_{24,24}; a_{1,25}; a_{2,26}; a_{4,28}; a_{5,29}; a_{6,30}; a_{9,27}; a_{10,28}; a_{11,29}; a_{12,30}; a_{20,26}; a_{25,27}; a_{26,25}; a_{3,31}; a_{7,32}; a_{29,27}; a_{30,25}; a_{30,27}; a_{32,25}; a_{27,33}; a_{28,34}; a_{29,34}; a_{31,33}; a_{31,34}; a_{32,33}$
$0.5 \cdot dc_x$	$a_{8,30}; a_{9,29}; a_{20,30}; a_{21,29}$
$0.5 \cdot dc_z$	$a_{2,28}; a_{14,28}$
$-0.5 \cdot dc_z$	$a_{1,29}; a_{13,29}$

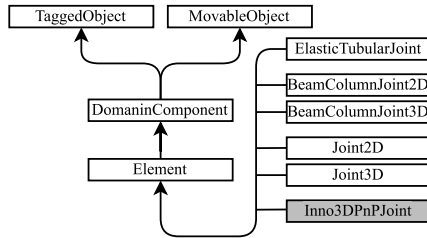


Fig. 17. Class diagram for the beam-to-column joint finite elements.

from which the stiffness matrix $\mathbf{K}_j(34 \times 34)$ of the joint finite element is obtained:

$$\mathbf{K}_j = \mathbf{A}^T \cdot \mathbf{k}_j \cdot \mathbf{A} \quad (18)$$

Since no external forces will be considered in the internal DOFs, the stiffness matrix \mathbf{K}_j can be statically condensed to be explicitly expressed in terms of the displacements of the external nodes. Therefore, the vector of nodal forces (Eq. (11)) and vector of nodal displacements (Eq. (14)) are split into 2 sub-vectors as follows:

$$\mathbf{P} = [\mathbf{P}_{\text{ext}} \quad \mathbf{P}_{\text{int}}]^T \quad (19)$$

$$\mathbf{U} = [\mathbf{U}_{\text{ext}} \quad \mathbf{U}_{\text{int}}]^T \quad (20)$$

where \mathbf{P}_{ext} and \mathbf{U}_{ext} are (30×1) , while \mathbf{P}_{int} and \mathbf{U}_{int} are (4×1) .

Therefore, Eq. (17) yields into:

$$\mathbf{K}_j \mathbf{U} = \begin{bmatrix} \mathbf{K}_{\text{ext}} & \mathbf{K}_{\text{ie}} \\ \mathbf{K}_{\text{ei}} & \mathbf{K}_{\text{int}} \end{bmatrix} \begin{bmatrix} \mathbf{U}_{\text{ext}} \\ \mathbf{U}_{\text{int}} \end{bmatrix} \quad (21)$$

where \mathbf{K}_{ext} is (30×30) , \mathbf{K}_{int} is (4×4) , \mathbf{K}_{ie} is (30×4) and \mathbf{K}_{ei} is (4×30) , from which the condensed stiffness matrix, $\mathbf{K}_j^{\text{cond}}(30 \times 30)$, is derived:

$$\mathbf{K}_j^{\text{cond}} = \mathbf{K}_{\text{ext}} - \mathbf{K}_{\text{ie}} \cdot (\mathbf{K}_{\text{int}})^{-1} \cdot \mathbf{K}_{\text{ei}} \quad (22)$$

5.3. Implementation in OpenSees

The proposed beam-to-column joint finite element is implemented in the architecture of OpenSees as the Inno3DPnPJoint Class [48], as presented in Fig. 17. The detailed implementation and the list of methods comprising the Inno3DPnPJoint Class is provided in Appendix A. Additionally, the user manual for the joint finite element is given in Appendix B.

6. Tube components stiffness

6.1. Equivalent frame-models

The mechanical properties of the components of the P&PJ are computed using some of the already available models in the Eurocode 3, Part 1-8 [5] in conjunction with other models for stiffness developed in the scope of Inno3Djoints project [39].

Although the stiffness of the tube components (k_F and k_I) can be computed using higher-order finite element models, as shown in

Section 4.2, or derived from experimental results, these procedures are time-consuming and impractical for design purposes, thus the need for more straightforward methods.

Since a simple and straightforward procedure is desired for the computation of the stiffness of the components, *equivalent frame-models* (EFM) are used, also known in the research field of tubular structures as *ring model* [26,49]. The EFM are an attractive procedure because they allow for defining the stiffness of the components analytically.

The basic principle behind the EFM is to represent the behaviour of a tubular column in the beam-to-column joint region using a planar frame-model composed of Euler–Bernoulli elements representing the tube faces at the cross-section level. The forces and deformations on the tube faces, which originate from the beam/truss-girder elements of the structure, are represented by forces and displacements in the spans of the frame-elements. In these EFM, the frame-elements are fully connected at their ends, i.e. in the corners, to account for the interaction between the tube faces. The frame-model has pinned supports at the four corners; thus, the cross-section's overall distortion is not considered. To utilise the Euler–Bernoulli elements, the EFM concept requires the definition of the geometry and boundary conditions of the frame-model, as well as a bending stiffness parameter for the linear elements of the frame, named equivalent bending stiffness, EI .

According to the literature review, existing EFM are only suitable for square tubular cross-sections and do not account for the size of the 'loaded region' of the joint, e.g., the size of the socket in P&PJ, in the interaction between the column faces. To address these limitations, 2 new main EFM are presented, as follows:

- *Harada extended frame-model for rectangular cross-sections* (HR-EFM shown in Fig. 18(a)) – represents an expansion of the model initially developed by Harada [25,26] for square cross-section tubes to encompass rectangular cross-sections. However, by ensuring equal dimensions for both faces of the tube, i.e. $L_1 = L_2$ in Fig. 18(a), the initial formulation by Harada is achieved.
- *Proposed frame-model for rectangular cross-sections* (PR-EFM shown in Fig. 18(b)) – is a model that accounts for the size of the P&PJ and is applicable to both square and rectangular cross-section tubes.

In the P&PJ, the connection is defined as the contact region between the sockets and column (Fig. 6). However, in the PR-EFM formulation, the connection includes the contact region and the region between the sockets as a single, larger rigid region with a finite length, b , and an infinite bending stiffness, EI_∞ . This approach simplifies the process of accounting for the size of the sockets.

In Fig. 18, L_1 and L_2 denote the width and length of the tubes, respectively. The distances from the socket to the column's corner along the width and length are denoted by a and c , respectively. The thickness of the tube, t_c , is not represented graphically; however, it is considered the same on all four sides.

6.2. General procedure

The analytical computation of the stiffness of the tube components, k_F and k_I , based on the definition presented in Section 4 using the EFM requires first the computation of the stiffness matrix associated with the

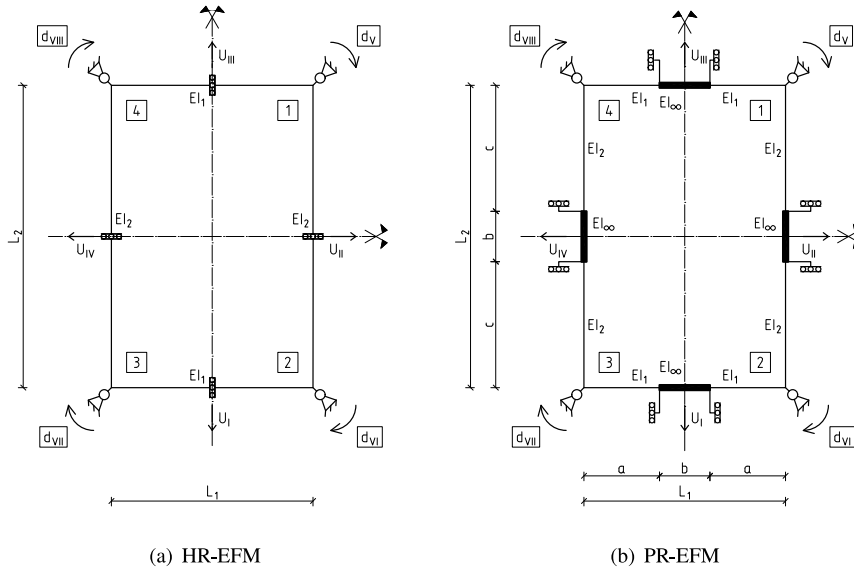


Fig. 18. Equivalent frame-models (EFM).

out-of-plane displacements at mid-span of each frame-element. This calculation can be done using various strategies; however, since the EFM are often highly redundant structures, the analytical determination of tube components may be more complex than desired.

Therefore, rather than calculating the full stiffness matrix of the EFM and condensing it to express it in terms of the out-of-plane translation at the frame-elements' mid-span, an alternative procedure was adopted. Following Harada [26], a procedure that involves the selection of four load patterns (LP-EFM) that, through symmetry considerations, allow for the easy computation of the out-of-plane mid-span displacements of the frame-elements to obtain the condensed stiffness matrix directly. These load patterns are shown in Fig. 19. The procedure involves solving a 16×16 linear system of equations corresponding to the condensed stiffness matrix of the EFM for the DOF corresponding to the out-of-plane deflections of the tube faces.

Taking advantage of the symmetry considerations for the LP-EFM, the calculation of the frame-element's mid-span deflection was performed on 1/4 of the EFM, referred to as the *simplified EFM* and shown in Fig. 20(b). For each simplified EFM, the mid-span deflection of the frame-elements is computed using the classic displacement method, which only requires the formulas for fixed-fixed and fixed-slide Euler-Bernoulli elements. The full calculation procedure consists of the following steps:

1. identification of the boundary conditions of the simplified EFM;
2. identification of the kinematic DOFs of the simplified EFM — in all cases, the simplified EFM has only one DOF when the fixed-fixed and fixed-slide Euler-Bernoulli elements are used: d ;
3. computation of the stiffness corresponding to the DOF: k_{LP-EFM_i} ;
4. computation of the displacement at the simplified EFM frame-elements' ends (mid-span deflections of the full EFM) caused by a unit displacement imposed on the DOF: $U_{LP-EFM_i}^{d=1}$;
5. computation of the fixation forces in the DOF d caused by the LP-EFM: r_{LP-EFM_i} ;
6. computation of the deflection at the frame-elements' ends caused by the LP-EFM when the DOF is restrained: $U_{LP-EFM_i}^P$;
7. computation of the displacement in the DOF associated with the LP-EFM: d_{LP-EFM_i} ;
8. computation of the displacement at the simplified EFM frame-elements' ends caused by the actual displacement: $U_{LP-EFM_i}^d$;
9. computation of the displacement at the simplified EFM frame-elements' ends caused by the LP-EFM using the principle of superposition of effects: U_{LP-EFM_i} .

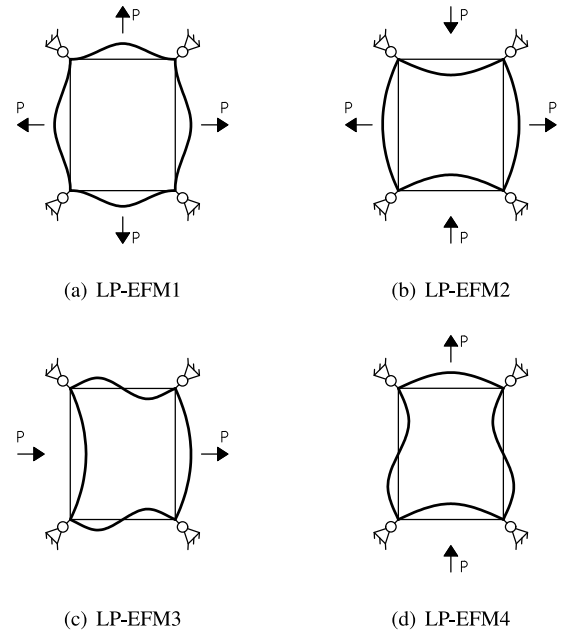


Fig. 19. Load patterns used to compute the stiffness matrix of EFM.

After obtaining the final displacement for every LP-EFM, the matrix F_{EFM} is constructed, inverted, and multiplied with the matrix of external forces, P_{EFM} , to obtain the condensed stiffness matrix of the EFM, K_{EFM} .

A worked example is provided below for the HR-EFM under LP-EFM2. In this example, the deflection, simplified frame, stiffness coefficients, and restraining forces are shown from Figs. 20(a) to 20(d).

The simplified EFM has only one DOF, i.e. rotation: d (Fig. 20(c)); thus, the stiffness is:

$$k_{LP-EFM2} = \frac{2 E I_1}{L_1} + \frac{2 E I_2}{L_2} \quad (23)$$

where I_1 and I_2 are the fictitious second moments of inertia of the top and right frame-elements of the EFM (Section 6.3).

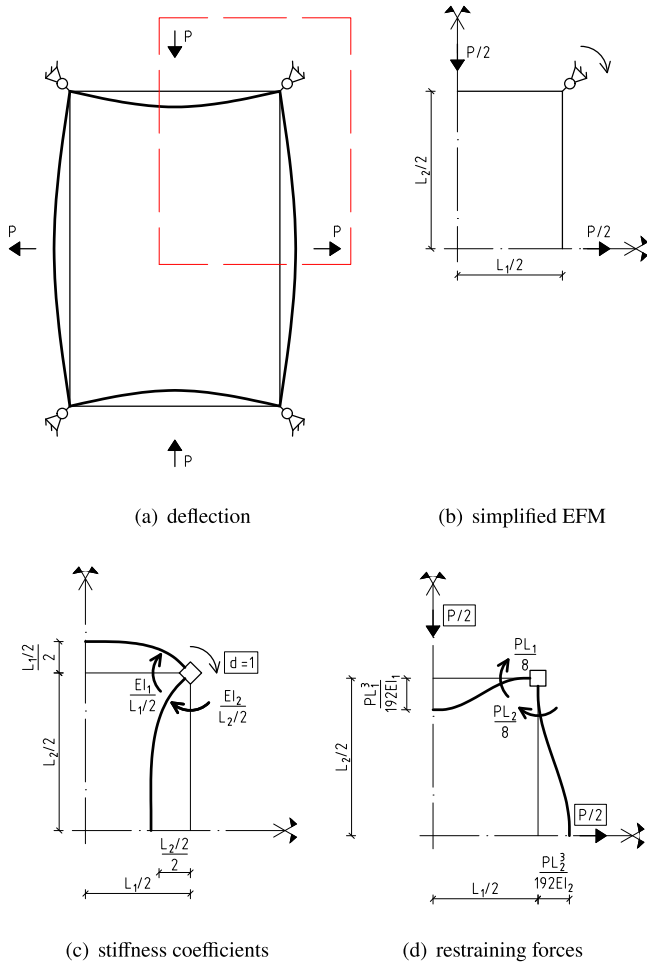


Fig. 20. HR-EFM: under LP-EFM2.

The out-of-plane displacements of column faces are:

$$\mathbf{U}_{LP-EFM2}^{d=1} = \begin{bmatrix} U_{III}^{d=1} \\ U_{II}^{d=1} \end{bmatrix} = \begin{bmatrix} \frac{L_1}{4} \\ -\frac{L_2}{4} \end{bmatrix} \quad (24)$$

where U_{III} and U_{II} are the out-of-plane displacements along coordinates III and II of the column caused by a unitary displacement, $d = 1$ (Fig. 18).

The forces required in the DOF d to have zero displacement, $d = 0$, (Fig. 20(d)) are:

$$r_{LP-EFM2} = \frac{P L_1}{8} + \frac{P L_2}{8} \quad (25)$$

and the corresponding out-of-plane displacements in the column faces are:

$$\mathbf{U}_{LP-EFM2}^P = \begin{bmatrix} U_{III}^P \\ U_{II}^P \end{bmatrix} = \begin{bmatrix} -\frac{P L_1^3}{192 E I_1} \\ \frac{P L_2^3}{192 E I_2} \end{bmatrix} \quad (26)$$

Accordingly, from statics, the displacement d caused by the LP-EFM2 in HR-EFM is:

$$d_{LP-EFM2} = (k_{LP-EFM2})^{-1} \cdot (-r_{LP-EFM2}) = -\frac{L_1 L_2 P (L_1 + L_2)}{16 E (I_1 L_2 + I_2 L_1)} \quad (27)$$

and the out-of-plane displacements of the column faces caused by the displacement $d_{LP-EFM2}$ becomes:

$$\mathbf{U}_{LP-EFM2}^d = \mathbf{U}_{LP-EFM2}^{d=1} \cdot d_{LP-EFM2} = \begin{bmatrix} -\frac{L_1^2 L_2 P (L_1 + L_2)}{64 E (I_1 L_2 + I_2 L_1)} \\ \frac{L_1 L_2^2 P (L_1 + L_2)}{64 E (I_1 L_2 + I_2 L_1)} \end{bmatrix} \quad (28)$$

Finally, the out-of-plane displacements of the column faces caused by LP-EFM2 on the HR-EFM are:

$$\mathbf{U}_{LP-EFM2} = \mathbf{U}_{LP-EFM2}^d + \mathbf{U}_{LP-EFM2}^P = \begin{bmatrix} U_{III}^{LP-EFM2} \\ U_{II}^{LP-EFM2} \end{bmatrix} = \begin{bmatrix} -\frac{L_1^2 P (I_2 L_1^2 + 4 I_1 L_1 L_2 + 3 I_1 L_2^2)}{192 E I_1 (I_1 L_2 + I_2 L_1)} \\ \frac{L_2^2 P (3 I_2 L_1^2 + 4 I_2 L_1 L_2 + I_1 L_2^2)}{192 E I_2 (I_1 L_2 + I_2 L_1)} \end{bmatrix} \quad (29)$$

The out-of-plane displacements of the other column faces, namely for coordinates I and IV, i.e. the bottom and left frame-elements, can be calculated similarly due to the double symmetry conditions.

After repeating this process for all LP-EFM depicted in Fig. 19, i.e. four times, the load matrix, \mathbf{P}_{EFM} is constructed for the full EFM. This load matrix is a concatenation of the load vectors of each LP-EFM, $\mathbf{P}_{LP-EFMi}$, which are constructed in a counterclockwise direction starting from the bottom face. Thus,

$$\mathbf{P}_{EFM} = \begin{bmatrix} \mathbf{P}_{LP-EFM1} & \mathbf{P}_{LP-EFM2} & \mathbf{P}_{LP-EFM3} & \mathbf{P}_{LP-EFM4} \end{bmatrix} = \begin{bmatrix} P & -P & 0 & -P \\ P & P & P & 0 \\ P & -P & 0 & P \\ P & P & -P & 0 \end{bmatrix} \quad (30)$$

Similarly, the matrix, \mathbf{F}_{EFM} , is a concatenation of the out-of-plane displacements associated with each LP-EFM, i.e. column i contains the displacement from LP-EFM i :

$$\mathbf{F}_{EFM} = \begin{bmatrix} U_{III}^{LP-EFM1} & U_{III}^{LP-EFM2} & U_{III}^{LP-EFM3} & -U_{III}^{LP-EFM4} \\ U_{II}^{LP-EFM1} & U_{II}^{LP-EFM2} & U_{II}^{LP-EFM3} & U_{II}^{LP-EFM4} \\ U_{III}^{LP-EFM1} & U_{III}^{LP-EFM2} & U_{III}^{LP-EFM3} & U_{III}^{LP-EFM4} \\ U_{II}^{LP-EFM1} & U_{II}^{LP-EFM2} & -U_{II}^{LP-EFM3} & U_{II}^{LP-EFM4} \end{bmatrix} \quad (31)$$

Lastly, the stiffness matrix for the HR-EFM, \mathbf{K}_{EFM} , is calculated:

$$\mathbf{K}_{EFM} = \mathbf{P}_{EFM} \cdot (\mathbf{F}_{EFM})^{-1} = \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} \\ k_{21} & k_{22} & k_{23} & k_{24} \\ k_{31} & k_{32} & k_{33} & k_{34} \\ k_{41} & k_{42} & k_{43} & k_{44} \end{bmatrix} \quad (32)$$

from which the stiffness coefficients are obtained:

$$k_{11} = k_{33} = \frac{48 E I_1 (I_1 L_2 + 4 I_2 L_1)}{L_1^3 (I_1 L_2 + I_2 L_1)} \quad (33)$$

$$k_{22} = k_{44} = \frac{48 E I_2 (4 I_1 L_2 + I_2 L_1)}{L_2^3 (I_1 L_2 + I_2 L_1)} \quad (34)$$

$$k_{12} = k_{14} = k_{21} = k_{23} = k_{32} = k_{34} = k_{41} = k_{43} = \frac{72 E I_1 I_2}{L_1 L_2 (I_1 L_2 + I_2 L_1)} \quad (35)$$

$$k_{13} = k_{24} = k_{31} = k_{42} = 0 \quad (36)$$

Furthermore, by applying the same procedure to the PR-EFM, the stiffness coefficients become:

$$k_{11} = k_{33} = \frac{6 E I_1 (4 I_2 a + I_1 c)}{a^3 (I_2 a + I_1 c)} \quad (37)$$

$$k_{22} = k_{44} = \frac{6 E I_2 (I_2 a + 4 I_1 c)}{c^3 (I_2 a + I_1 c)} \quad (38)$$

$$k_{12} = k_{14} = k_{21} = k_{23} = k_{32} = k_{34} = k_{41} = k_{43} = \frac{9 E I_1 I_2}{a c (I_2 a + I_1 c)} \quad (39)$$

$$k_{13} = k_{24} = k_{31} = k_{42} = 0 \quad (40)$$

The stiffness of the column components (k_F and k_I) can be obtained by setting the stiffness matrix given by Eq. (4) equal to the stiffness matrix given by Eq. (32). Accordingly, the stiffnesses of the tube components for the HR-EFM become:

$$k_{F, HR-EFM}^{AC} = \frac{48 E I_1 (-3 I_2 L_1^2 + 4 I_2 L_1 L_2 + I_1 L_2^2)}{L_1^3 L_2 (I_1 L_2 + I_2 L_1)} \quad (41)$$

$$k_{F, HR-EFM}^{BD} = \frac{48 E I_2 (I_2 L_1^2 + 4 I_1 L_1 L_2 - 3 I_1 L_2^2)}{L_1 L_2^3 (I_1 L_2 + I_2 L_1)} \quad (42)$$

$$k_{I, HR-EFM} = \frac{72 E I_1 I_2}{L_1 L_2 (I_1 L_2 + I_2 L_1)} \quad (43)$$

and for the PR-EFM are:

$$k_{F, PR-EFM}^{AC} = \frac{6 E I_1 (-3 I_2 a^2 + 4 I_2 a c + I_1 c^2)}{a^3 c (I_2 a + I_1 c)} \quad (44)$$

$$k_{F, PR-EFM}^{BD} = \frac{6 E I_2 (I_2 a^2 + 4 I_1 a c - 3 I_1 c^2)}{a c^3 (I_2 a + I_1 c)} \quad (45)$$

$$k_{I, PR-EFM} = \frac{9 E I_1 I_2}{a c (I_2 a + I_1 c)} \quad (46)$$

6.3. Second moment of inertia of 1D elements for the EFM

The mechanical properties for the EFM are based on the work of Neves and Gomes [50], and Simões da Silva et al. [51] on the out-of-plane behaviour of the column's web plate in beam-to-minor-axis column joints. The model developed by Neves and Gomes (NGM) focuses on the out-of-plane force–deformation relation curve for a rectangular plate with two opposite sides fixed and a central region loaded by a rigid-rectangular plate, as shown in Fig. 21. According to this model, the initial stiffness, S_i , of the out-of-plane force–deformation relation is:

$$S_i = \frac{16 E t_c^3}{L^2} \frac{\alpha + (1 - \beta) \tan \theta}{(1 - \beta)^3 + \frac{10.4 (k_1 - k_2 \beta)}{\mu^2}} \quad (47)$$

where:

f – width of the rigid plate;

u – height of the rigid plate;

L – width of the plate;

t_c – thickness of the plate;

$$\beta = \frac{f}{L}$$

$$\alpha = \frac{u}{L}$$

$$\theta = 35 - 10 \beta \text{ (in degrees)}$$

$$\mu = \frac{L}{t_c}$$

$$k_1 = 1.5$$

$$k_2 = 1.63$$

The NGM is valid under the following geometrical conditions:

$$\begin{aligned} 0 &\leq \alpha \leq 0.2 \\ 0.08 &\leq \beta \leq 0.8 \\ 10 &\leq \mu \leq 50 \end{aligned} \quad (48)$$

Although the NGM focuses on the out-of-plane behaviour of a single-column plate, it is used in the current research due to the absence of

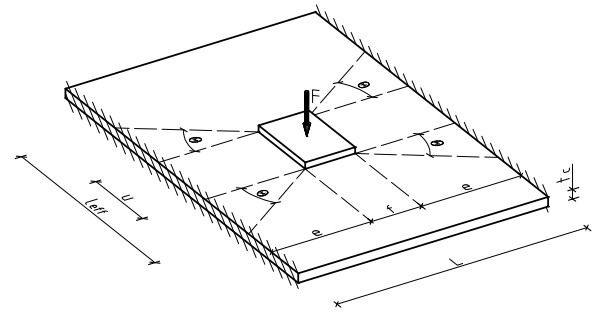


Fig. 21. Neves-Gomes Model (NGM). Source: Adapted from [50]

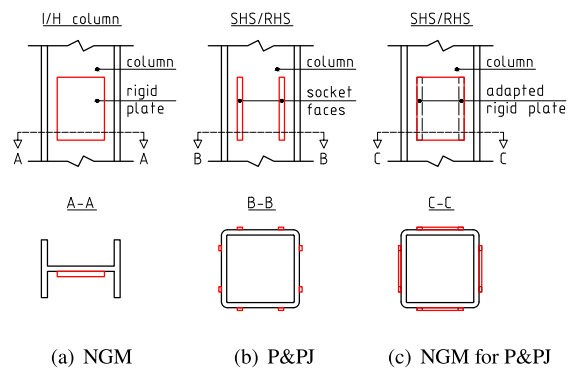


Fig. 22. Neves-Gomes Model adapted to innovative plug-and-play joints (schematic representation).

a more suitable formulation. As a result, the tube faces are considered as four isolated column plates interacting with each other. However, applying this methodology to the P&PJ results in some limitations; for example, the region between the socket faces is considered rigid, as shown in Fig. 22.

The bending stiffness is determined so that the deflection of the frame-elements matches the out-of-plane deformation of the column plate. Therefore, the deflection, δ_1 , caused by a load, P , on an effective strip of size $f \times l_{eff}$, positioned centrally as in Fig. 21, and with a second moment of inertia of I , is computed considering a 1D element with fixed BCs at both ends:

$$\delta_1 = \frac{P e^3}{24 E I} \quad (49)$$

In the NGM, the out-of-plane deformation, δ_2 , of an isolated single column plate is obtained from the initial stiffness, S_i :

$$\delta_2 = \frac{P}{S_i} \quad (50)$$

By setting the right-hand side of Eq. (49) equal to the right-hand side of Eq. (50), the bending stiffness of the frame-elements becomes:

$$\delta_1 = \delta_2 \rightarrow EI = \frac{e^3}{24} S_i \quad (51)$$

Following the same principle, the second moment of inertia of the frame-elements of the HR-EFM are:

$$I_{1, HR-EFM} = \frac{L_1^3}{192 E} S_i \quad (52)$$

$$I_{2, HR-EFM} = \frac{L_2^3}{192 E} S_i \quad (53)$$

and of the PR-EFM are:

$$I_{1, PR-EFM} = \frac{a^3}{24 E} S_i \quad (54)$$

$$I_{2,PR-EFM} = \frac{c^3}{24 E} S_i \quad (55)$$

To enhance the computational process for determining the initial stiffness (Eq. (47)), the second moment of inertia for the frame-elements (Eqs. (52) to (55)), and the stiffness properties of tube components (Eqs. (41) to (46)) in the context of the Inno3DPnJoint beam-to-column finite element a dedicated Jupyter notebook [52] was developed. This notebook is publicly accessible on GitHub [53], allowing for dynamic launching via Binder [54]. Furthermore, a video tutorial is available on YouTube, providing instructions on utilising the notebook effectively [55]. Consequently, users have the capability to adjust parameters, replicate calculations, and generate configurations and models tailored to their research objectives.

7. Validation

The validation of the beam-to-column joint finite element implemented in OpenSees [7] was carried out by comparing it to RFEMs built in Abaqus [34] using the procedure described in Section 4.2.2. The results are presented in terms of the out-of-plane displacements of each column face individually.

The validation models were built varying the following aspects:

1. cross-sections: 2, namely SHS and RHS (Fig. 7(a));
2. LP: 10 for SHS (Fig. C.1) and 17 for RHS (Fig. C.2);
3. SC: 4, with L_{SC} equal to 0 mm, 40 mm, 90 mm and 140 mm (Fig. 7(b));
4. EFM: 2, namely HR-EFM and PR-EFM (Fig. 18); the EFM for the square cross-section is obtained by equating the length and width in the rectangular formulation, i.e. $L_1 = L_2$ and $a = c$;
5. since the rounded corners of the tubular element could lead to uncertainty when it comes to choosing the correct width of the column's plate, i.e. L in Fig. 21, which is needed in the computation of k_F and k_I for the EFM, 2 cases were analysed:
 - (a) R0 – when the length of the column plate is equal to the distance between the exterior of any two parallel column faces, i.e. the radius is neglected, thus, $L = 200$ mm, and $L_1 = 200$ mm and $L_2 = 300$ mm for the SHS and RHS, respectively;
 - (b) R1 – when the length of the column plate is equal to the distance between the radii of a face, i.e. the radius is considered, thus, $L = 177$ mm, and $L_1 = 177$ mm and $L_2 = 277$ mm for the SHS and RHS, respectively;
6. the RHS presents 2 sub-configurations based on the moment of inertia:
 - (a) IF – when each column face is assigned its corresponding moment of inertia, i.e. I_1 and I_2 ;
 - (b) IEq – when each column face is assigned an average moment of inertia, i.e. $I_{eq} = \text{avg}(I_1, I_2)$.

The aforementioned aspects lead to a pool of 160 models for the SHS (i.e. 2 EFM \times 2 radius \times 10 LP \times 4 SC) and 544 models for the RHS (i.e. 4 EFM \times 2 radius \times 17 LP \times 4 SC). As the main objective is to evaluate the 3D interaction between the column faces, the connection components are assumed to have infinite stiffness.

The deformation of SHS and RHS cross-sections for all LP with 'SC02' is presented in Figs. C.3 and C.4, respectively.

The out-of-plane displacements of the column faces between the Abaqus and OpenSees models are compared using the standard deviation, SD , and coefficient of determination, R^2 , defined by Eqs. (56) and (57), respectively.

$$SD = \text{std} \left(\frac{\text{disp}_{\text{OpenSees}}}{\text{disp}_{\text{Abaqus}}} \right) \quad (56)$$

Table 7
 R^2 for SHS.

EFM	Radius	Face A	Face B	Face C	Face D
HS-EFM	R0	0.86	0.88	0.70	0.70
PS-EFM	R0	0.86	0.88	0.70	0.70
HS-EFM	R1	0.97	0.97	0.92	0.94
PS-EFM	R1	0.97	0.97	0.92	0.94

Table 8
 SD for SHS.

EFM	Radius	Face A	Face B	Face C	Face D
HS-EFM-IF	R0	0.22	0.43	0.31	0.61
PS-EFM-IF	R0	0.22	0.43	0.31	0.61
HS-EFM-IF	R1	0.27	0.34	0.31	0.46
PS-EFM-IF	R1	0.27	0.34	0.31	0.46

Table 9
 R^2 for RHS.

EFM	Radius	Face A	Face B	Face C	Face D
HR-EFM-IF	R0	0.73	0.85	0.75	0.85
HR-EFM-IEq	R0	0.75	0.83	0.77	0.83
PR-EFM-IF	R0	0.74	0.81	0.76	0.81
PR-EFM-IEq	R0	0.80	0.75	0.82	0.77
HR-EFM-IF	R1	0.95	0.96	0.94	0.96
HR-EFM-IEq	R1	0.95	0.93	0.95	0.94
PR-EFM-IF	R1	0.94	0.97	0.93	0.97
PR-EFM-IEq	R1	0.94	0.96	0.92	0.96

Table 10
 SD for RHS.

EFM	Radius	Face A	Face B	Face C	Face D
HR-EFM-IF	R0	0.58	0.37	1.87	0.59
HR-EFM-IEq	R0	0.58	0.44	1.76	0.62
PR-EFM-IF	R0	1.46	0.36	2.18	1.35
PR-EFM-IEq	R0	1.65	0.36	2.20	1.60
HR-EFM-IF	R1	0.59	0.32	0.96	0.59
HR-EFM-IEq	R1	0.47	0.39	0.73	0.54
PR-EFM-IF	R1	1.43	0.29	2.01	1.47
PR-EFM-IEq	R1	1.50	0.30	1.85	1.62

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \quad (57)$$

where SS_{res} and SS_{tot} represent the residual sum of squares and the total sum of squares, i.e. proportional to the variance of the data, respectively. The 'ideal' case is characterised by a coefficient of determination equal to 1, $R^2 = 1$, indicating that the regression line fitting the data is linear and given by:

$$y = m x + b, \text{ with } m = 1 \text{ and } b = 0 \quad (58)$$

The standard deviation and coefficient of determination are presented for the SHS in Tables 7 and 8, and for the RHS in Tables 9 and 10. The calculations are performed for each EFM at the face level.

The coefficients of determination from Tables 7 and 9 show that a higher degree of correlation is achieved in the R1 case for both SHS and RHS. Additionally, in the SHS case, both HS-EFM and PS-EFM produce the same results, indicating that the size of the joint, i.e. socket, does not affect the results (Table 7). For the RHS case, a

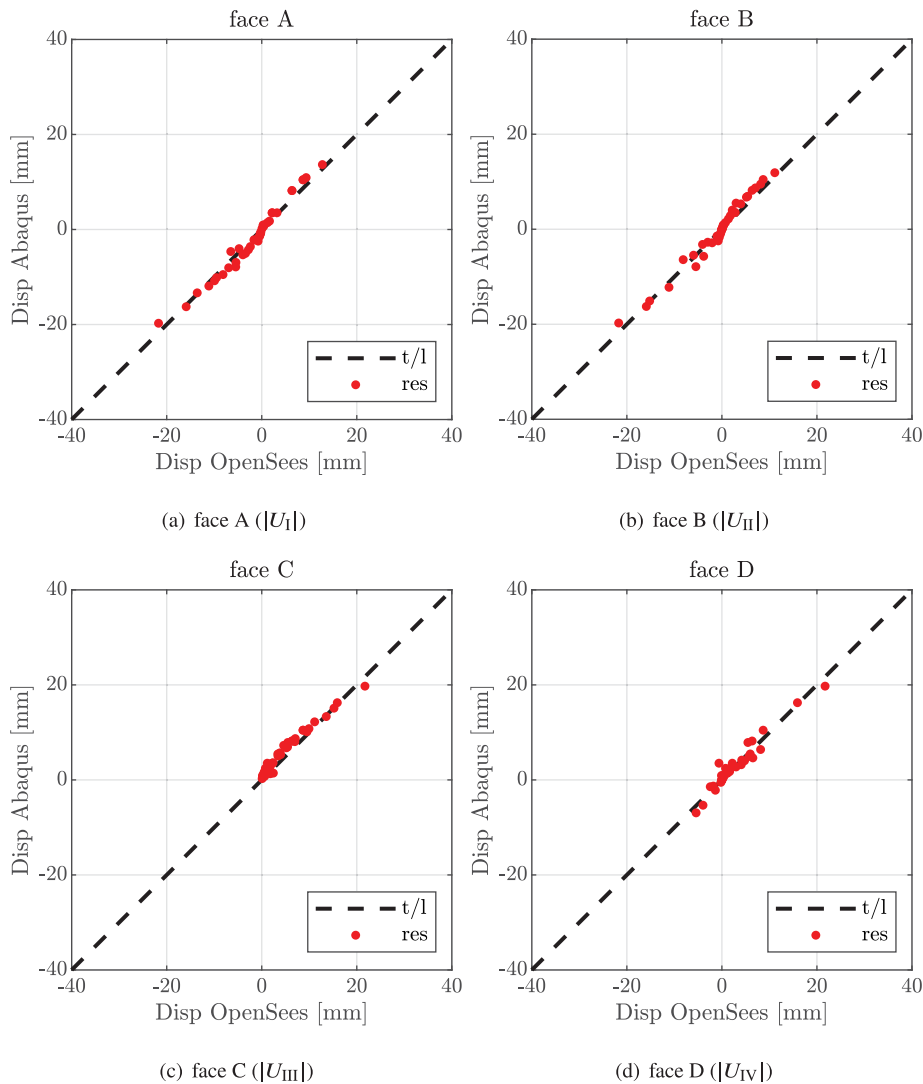


Fig. 23. PS-EFM-R1.

higher degree of correlation is observed when using IEq with R0, and IF with R1 (Table 9). These conclusions are also evident from the standard deviation.

The out-of-plane displacements extracted from the centre point of each column face from both Abaqus and OpenSees are plotted against each other in Figs. 23 and 24 for PS-EFM-R1 and PR-EFM-IF-R1, respectively. In an ideal case, in which there is equality between the displacements from the two software, the results, i.e. red dots: res , would be distributed along a line that passes through the origin and has a unit gradient, $y = x$, i.e. black dashed line: t/l . Although this is not the case, the displacements obtained from OpenSees using the beam-to-column joint finite element are in good agreement with those obtained from Abaqus.

8. Conclusions

This research presented the comprehensive development, implementation, and validation of a novel macro-element developed especially for the innovative three-dimensional plug-and-play joints between tubular columns and lightweight steel truss-girders proposed within the Inno3DJoints project [1]. In addition, this macro-element proves highly suitable for the linear analysis conducted in everyday design practices. Moreover, the macro-element is based on the component

method, whereby its stiffness and geometric parameters have a clear, physical, and intuitive meaning. Furthermore, the stiffness parameters for the model can be obtained from higher-order finite element models, experimental tests, or equivalent frame-models.

In this research, equivalent frame-models were developed for SHS and RHS cross-sections with consideration given to the size of the joint. As there is currently no better alternative, the Neves-Gomes Model was used to derive analytical expressions for the stiffness parameters required for the macro-element. This assumption should be revisited once a more straightforward method for calculating the mechanical properties of the innovative plug-and-play joints becomes available.

The concept of the macro-element was validated against higher-order finite element models developed in Abaqus and was subsequently implemented as a standalone beam-to-column joint finite element in the OpenSees framework. The Inno3DPnPJoint command can be used to generate this element.

A comprehensive parametric study was conducted to evaluate the beam-to-column joint finite element outputs when its required stiffness parameters are determined using equivalent frame-models. The results obtained using the joint finite element in OpenSees were compared to those from higher-order finite element models developed in Abaqus and showed good agreement.

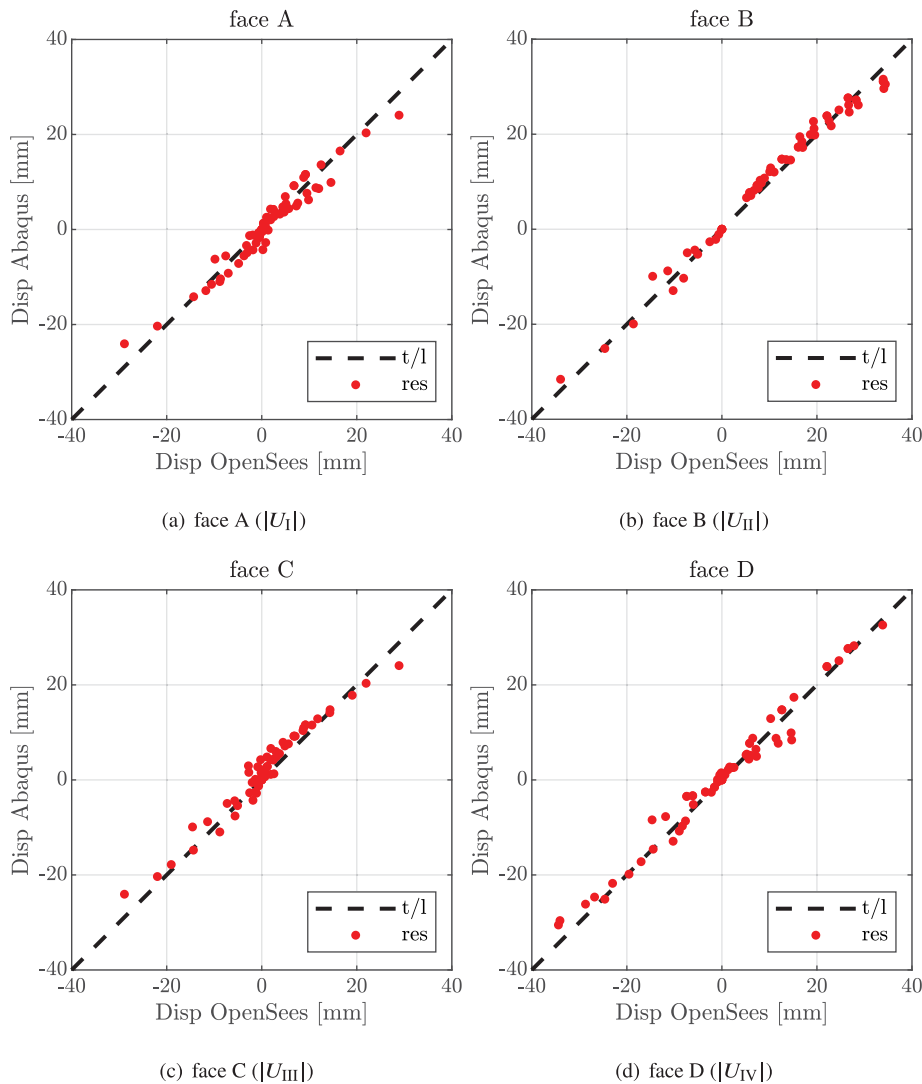


Fig. 24. PR-EFM-IF-R1.

CRedit authorship contribution statement

Cristian V. Miculaş: Methodology, Software, Validation, Writing – original draft, Conceptualization. **Ricardo J. Costa:** Conceptualization, Software, Validation, Writing – original draft. **Luis Simões da Silva:** Validation, Supervision, Writing – review & editing. **Rui Simões:** Validation, Writing – review & editing. **Helder Craveiro:** Validation, Writing – review & editing. **Trayana Tankova:** Validation, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

This work was partly financed by:

1. FCT/MCTES through national funds (PIDDAC) under the R&D Unit Institute for Sustainability and Innovation in Structural Engineering (ISISE), under reference UIDB/04029/2020;
2. Research Fund for Coal and Steel under grant agreement No. 749959 (INNO3DJOINTS);
3. national funds through FCT – Foundation for Science and Technology, Portugal, under grant agreement PTDC/ECI-EGC/31834/2017 (INNO3DJOINTS);
4. national funds through FCT – Foundation for Science and Technology, Portugal, under grant agreement SFRH/BD/138151/2018 attributed to Cristian V. Miculaş.

Appendix A. Inno3DPnPJoint: implementation in OpenSees

The Inno3DPnPJoint Class is a child of the base abstract Element Class, as presented in Fig. 17. The Element Class consists of several virtual methods defined by the children of the class. These methods dynamically allocate and deallocate memory for an instance of the class, initialise an instance of the class, perform inquiry and access tasks, and generate and return an element tangent matrix and residual vector, as described in [28].

The programming language used to code the proposed beam-to-column joint is C++ [56]. The integrated development environment

used to compile the OpenSees source code is Microsoft VisualStudio 2019 [57].

For the development of any new finite element ideas, two files must be coded: a header file (*.h) and a main (implementation) file (*.cpp). The former file defines the interface and variables for the new class, which should be a subclass of the Element Class, while the latter file explains in detail the functions of the constructors, destructor, and other methods.

The proposed beam-to-column joint element requires an internal solution to determine the internal nodal displacements that satisfy the internal equilibrium of the element. In addition, the presence of the internal DOFs in the proposed model and the nonlinear behaviour of the components enforce an internal iteration scheme to be implemented at compilation time, i.e. not at run time.

A.1. Transformation matrix

To remove any coordinate-/sign-related restrictions for the user in defining the joint, i.e. to allow the user to define the beam-to-column joint element in any desired position in the 3D space, a transformation matrix is added to the code through the formTransfMat() method (Table A.1). This matrix, **Transf**(30 × 30), is derived from the vectors containing the coordinates for nodes 1 and 2 and the coordinates of their cross-product. First, a verification is made to ensure that the norms of these three vectors are different from 0. Then, a transformation sub-matrix, **Tran**(3 × 3), is constructed by dividing the coordinates of the three vectors by their norms. Finally, the transformation matrix **Transf** is obtained by assembling the sub-matrix **Tran** 10 times diagonally.

The transformation matrix **Transf** is then used in the following methods presented in detail in Table A.1

- getGlobalDispls(...) – in the OpenSees input file, i.e. *.tcl or *.py, the joint element is defined in the global coordinate system; therefore, the coordinates of the five nodes defining the joint element are transformed into the local coordinate system;
- formR(...) – the local residual force vector of the element, contains the values related to both the external and internal DOFs; thus, the ones associated with the external DOFs are transformed back, from the local to the global coordinate system;
- formK(...) – the condensed local stiffness matrix of the element, is transformed from the local to the global coordinate system;
- getResponse(...) – the committed displacements associated with the external DOFs are transformed back from the local to the global coordinate system.

A.2. Element class

The methods comprising the Inno3DPnPJoint Class are listed in Table A.1 and are adapted from [28].

Appendix B. Inno3DPnPJoint user manual

B.1. Element info

The proposed beam-to-column joint finite element can be viewed as a 2D plate in a 3D space defined by five external nodes (Fig. 16), each having six DOFs. This element configuration is suitable for rectangular and square cross-sections. Furthermore, it allows for the simulation of all types of joints: central, edge, and corner, with some nodes, potentially not being connected to any beam/truss-girder or column elements.

B.2. Command line and input arguments

The Inno3DPnPJoint beam-to-column joint element can be constructed using the command input lines given below with the input arguments listed in Table B.1.

- cmd input line for *.tcl files:

```
element Inno3DPnPJoint $eleTag <$Node1 $Node2
$Node3 $Node4 $Node5> <$SprMatTag01 ...
$SprMatTag32>
```

cmd Example:

```
element Inno3DPnPJoint 99 101 102 103 104 105 1 2 3 4
5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
26 27 28 29 30 31 32

# constructs an Inno3DPnPJoint joint element with ele-
ment tag 99, that is connected to nodes 101, 102, 103, 104
and 105 and uses for the components' behaviour the uniaxial
material object tags from 1 to 32.
```

- cmd input line for *.py files:

```
element('Inno3DPnPJoint', $eleTag, <$eleNode1,
$eleNode2, $eleNode3, $eleNode4>, <$SprMatTag01,
..., $SprMatTag32>)
```

B.3. Node definition requirements

The nodes must be defined in a specific order and have coordinates that meet the requirements of coplanarity, perpendicularity, collinearity, centrality, and dimensionality to successfully create the beam-to-column joint finite element. These requirements ensure that the coordinates of the five nodes given as input by the user create a 2D plane in 3D space. These checks can be found in the setDomain method (Table A.1).

B.3.1. Input order

The nodes should be defined in a counterclockwise direction. For example, if the element is created in the X-Z plane, the node's input order should be as follows: 1/bottom, 2/right, 3/top, 4/left, and 5/centre, as shown in Fig. 16. However, the joint can be defined in any plane if the node order definition is adapted accordingly.

B.3.2. Coplanarity

The equation of the plane passing through nodes 1, 2, and 3 is calculated to ensure that the nodes lie in the same plane. Then, a check is performed to see if nodes 4 and 5 satisfy this equation. If the

Table A.1
Methods Comprising the Inno3DPnPJoint Class ([PU]– Public, [PR]– Private).

Method name	Description
<i>Constructor</i>	
[PU] Inno3DPnPJoint(...)	Initializes an object of the class.
<i>Destructor</i>	
[PU] Inno3DPnPJoint()	Performs dynamic storage deallocation.
<i>Inquiry and access methods</i>	
[PU] virtual int getNumExternalNodes()	Returns the number of external nodes of the element.
[PU] virtual const ID &getExternalNodes()	Returns pointers to the specified node tags.
[PU] virtual getNodePtrs()	Returns pointers to the specified node tags.
[PU] virtual int getNumDOF()	Returns the number of DOFs of the element.
[PU] virtual void setDomain()	Required for checking the DOFs and associativity with the node.
[PU] virtual const Matrix &getTangentStiff()	Returns the local stiffness matrix of the element at the global assembly stage for all the elements.
[PU] virtual getInitialStiff()	Returns the local stiffness matrix of the element at the global assembly stage for all the elements.
[PU] virtual const Vector &getResistingForce()	Returns the local residual force vector at the stage of global assembly.
[PU] virtual int getResponse(...)	Returns element specific responses.
[PU] virtual setResponse(...)	Contains the valid output(s) for the element.
[PU] virtual void Print(...)	Prints out element data.
[PU] virtual int displaySelf()	Displays element graphically.
<i>Element solution methods</i>	
[PR] void getGlobalDispls(...)	Returns a set of converged displacements for the five external nodes of the element.
[PR] void getBCJoint()	Returns the compatibility matrix.
[PR] void getdDef_du()	Extracts from the compatibility matrix the rows and columns related to internal DOFs.
[PR] void getdg_df()	Transposes the dDef_du matrix.
[PR] void matDiag(...)	Takes in the vector containing the stiffness of the components as input and converts it to a diagonal matrix of components.
[PR] void getMatResponse(...)	Takes in the displacements as input and returns the tangent and residual forces based on the material specified by the user.
[PR] void formR(...)	Forms the local residual force vector of the element.
[PR] void formK(...)	Forms the local stiffness matrix of the element.
[PR] void formTransfMat()	Forms the transformation matrix used to transform from global to local coordinates.
[PR] double getStepSize(...)	Determines the step size to be used for line search if there is a convergence problem with the internal equilibrium. This method increases the robustness of the element.
[PU] virtual int commitState()	Commits the displacement at each node once the internal equilibrium criteria for the element are met.
[PU] virtual int revertToLastCommit()	Returns to the last committed state if the analysis fails.
[PU] virtual int revertToStart()	Returns to the start if the analysis fails.
[PU] virtual int update()	Updates the displacements at the external nodes of the element.

Table B.1
Input arguments for Inno3DPnPJoint joint element.

Input \$args	Description
\$eleTag	An integer tag identifying the element in the domain
\$Node1	An integer tag indicating the node 1
...	
\$Node5	An integer tag indicating the node 5
\$SprMatTag01	An integer tag indicating the uniaxial materials for component 1
...	
\$SprMatTag32	An integer tag indicating the uniaxial materials for component 32

condition is met, all five nodes are coplanar, and the analysis continues. Contrarily, the analysis exits and displays this error message:

ERROR: Inno3DPnPJoint::setDomain – Node 4 does NOT belong to plane created by Node 1, Node 2 and Node 3. Check node coordinate definition.

or

ERROR: Inno3DPnPJoint::setDomain – Node 5 does NOT belong to plane created by Node 1, Node 2 and Node 3. Check node coordinate definition.

B.3.3. Perpendicularity

If the coplanarity requirement is fulfilled, the dot product of the vectors between nodes 1 and 3 and nodes 2 and 4 is calculated. If the result is 0, the two vectors are perpendicular, so the condition is

Table B.2
Common errors.

Description	Displayed error message
Number of input arguments is wrong, i.e. different than 38	WARNING error insufficient. arguments. Want: element Inno3DPnPJoint eleTag? Node1? Node2? Node3? Node4? Node5? Spring01? Spring02? ... Spring32?.
Number of external nodes is wrong, i.e. different than 5	ERROR: Inno3DPnPJoint::Inno3DPnPJoint() <i>eleTag</i> failed to create an ID of size 5.
Constructor fails to get copy of material and/or the copy is not valid	ERROR: Inno3DPnPJoint::Constructor failed to get a copy of material <i>matTag</i> .
Constructor fails to set pointer to external nodes	ERROR: Inno3DPnPJoint::setDomain. Node pointer is NULL. Node <i>nodeTag</i> does not exit in the domain.
The domain is null	ERROR: Inno3DPnPJoint::setDomain — Domain is null.
Number of DOFs is wrong, i.e. different than 6	ERROR: Inno3DPnPJoint::setDomain — number of DOF associated with the nodes is incorrect.
Number of spring requested as output is out of range (1–32)	ERROR: Inno3DPnPJoint::setResponse number of springs out of range: <i>springNo</i> . Spring numbers go from 1 to 32.

satisfied, and the analysis continues. Contrarily, the analysis exits and displays this error message:

ERROR: Inno3DPnPJoint::setDomain – vector of Node 1 & Node 3 not perpendicular to vector of Node 2 & Node 4. Check node coordinate definition.

B.3.4. Collinearity

The sets of nodes 1, 5, 3, and 2, 5, 4 need to be collinear, so four vectors are created between the centre node, i.e. node 5, and the edge nodes, i.e. nodes 1, 2, 3, and 4, and their cross-product is calculated. If the result is 0, the two sets of nodes are collinear, so the condition is

satisfied, and the analysis continues. Contrarily, the analysis exits and displays this error message:

ERROR: Inno3DPnPJoint::setDomain – Node 1, Node 5 and Node 3 are not collinear. Check node coordinate definition.

or

ERROR: Inno3DPnPJoint::setDomain – Node 2, Node 5 and Node 4 are not collinear. Check node coordinate definition.

B.3.5. Centrality and dimensionality

The edge nodes should be defined at the centre of each column face so the norms of the four vectors from the collinearity check are calculated and compared. There are three possible outcomes for this comparison: (i) all norms are equal, indicating that the tube is square, (ii) two sets of two equal norms are obtained, indicating that the tube is rectangular, and (iii) none of the previous conditions are met, in which case the analysis exits and displays an error message:

ERROR: Inno3DPnPJoint::setDomain – nodes are not located at the center of the column face. Check node coordinate definition.

In addition, the dimensions of the joint should be large enough to avoid division by 0. If this occurs, it is advised to change the units. To ensure this, the norms of the four vectors should be greater than a specific value set to $1e-3$. If this condition is met, the analysis continues. Contrarily, the analysis exits and displays this error message:

ERROR: Inno3DPnPJoint::setDomain – length or width $\leq 1e-3$, division by zero occurs. Increase joint size.

B.4. Common errors

There exists a multitude of error messages that may manifest due to various factors, supplementing those that arise from unmet node definition requirements. For illustrative purposes, Table B.2 presents a compilation of the most frequently encountered error messages.

B.5. Output recorders

B.5.1. Element level

The outputs at the element level can be requested using the following recorder commands:

Table B.3
Output \$args at element level.

Output \$args	C++ output variable		Description
	Name	Size	
extDisp extdisp	UeprCommit_G	30 × 1	Returns the displacement for the external DOFs.
intDisp intdisp	UeprIntCommit	4 × 1	Returns the displacement for the internal DOFs.
Disp disp	UeprCommit_G & UeprIntCommit	34 × 1	Returns the displacement for the external and internal DOFs.
Reaction reaction	R	34 × 1	Returns the global residual forces for all DOFs.
matStress matstress Stress stress	MaterialPtr->getStress()	32 × 1	Returns the stress values from the joint's components.
matStrain matstrain Strain strain	MaterialPtr->getStrain()	32 × 1	Returns the strain values from the joint's components.
matStressStrain matstressstrain StressStrain stressStrain	MaterialPtr->getStress() MaterialPtr->getStrain()	64 × 1	Returns the stress and strain values from the joint's components.

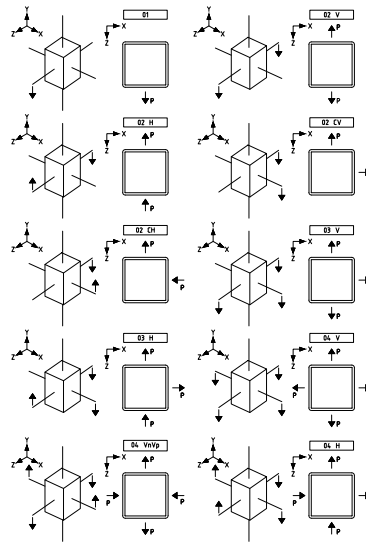


Fig. C.1. LPs for SHS.

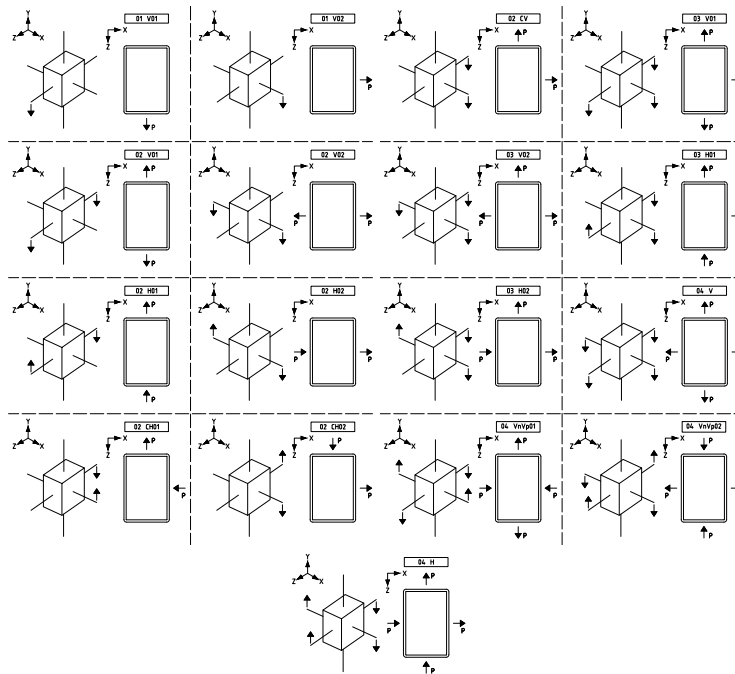


Fig. C.2. LPs for RHS.

- input line for *.tcl files:

```
recorder Element <-file $fileName> <-time> <-ele $ele-Tag> $arg
```

- input line for *.py files:

```
recorder('Element', '-file', 'fileName', '-time', '-ele', 'ele-Tag', '$arg')
```

The list of valid inputs for the argument, \$arg, at element level, is given by Table B.3.

B.5.2. Component level

The outputs at the component level can be requested using the following recorder commands:

Table B.4
Output \$arg at component level.

Output \$args	C++ output variable		Description
	Name	Size	
spring -spring material -material	MaterialPtr[springNo]	1 × 2	Returns a pair of stress-strain for each time step.

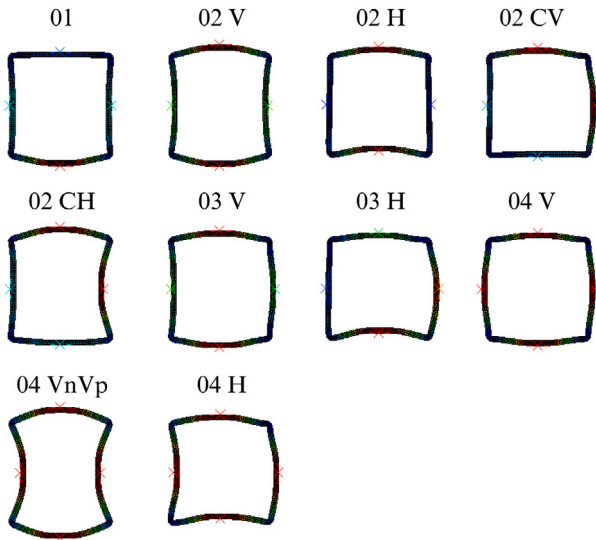


Fig. C.3. Cross-section deformation in W0 for SHS for all LPs with SC02.

- input line for *.tcl files:

```
recorder Element <-file $fileName> <-time> <-ele $ele-Tag> <$arg $sprNo stressStrain>
```

- input line for *.py files:

```
recorder('Element', '-file', 'fileName', '-time', '-ele', 'ele-Tag', '$arg', 'sprNo', 'stressStrain')
```

The list of valid inputs for the argument, \$arg, at the component level is given by Table B.4.

Appendix C. Load patterns and cross-section deformation

The load patterns (LPs) used throughout this research are illustrated in Figs. C.1 and C.2 for the SHS and RHS, respectively. These LPs represent gravitational and horizontal loading on corner, edge, and internal joints. In Figs. C.1 and C.2, each set of graphs includes: (i) on the left side, 3D representations of the tubular column and top chord of the CFS truss-girders (referred to as ‘Joint 1’ in Fig. 4(a)), along with the applied LP; and (ii) on the right side, 2D representations of the tube’s cross-section and the 2D equivalent of the applied 3D LP.

In addition, the deformations of SHS and RHS cross-sections for all LPs with ‘SC02’ are presented in Figs. C.3 and C.4, respectively.

Appendix D. Analytical and FE models mapping

The work done for the development of the beam-to-column joint finite element in this manuscript comprises two stages: ‘Stage 1: Validation of the CME Concept’ and ‘Stage 2: Validation of the Inno3DPnPJoint FE’. Fig. D.1 illustrates the mapping between the analytical and finite element models, including the associated software used for computation and their respective sections detailed in the article.

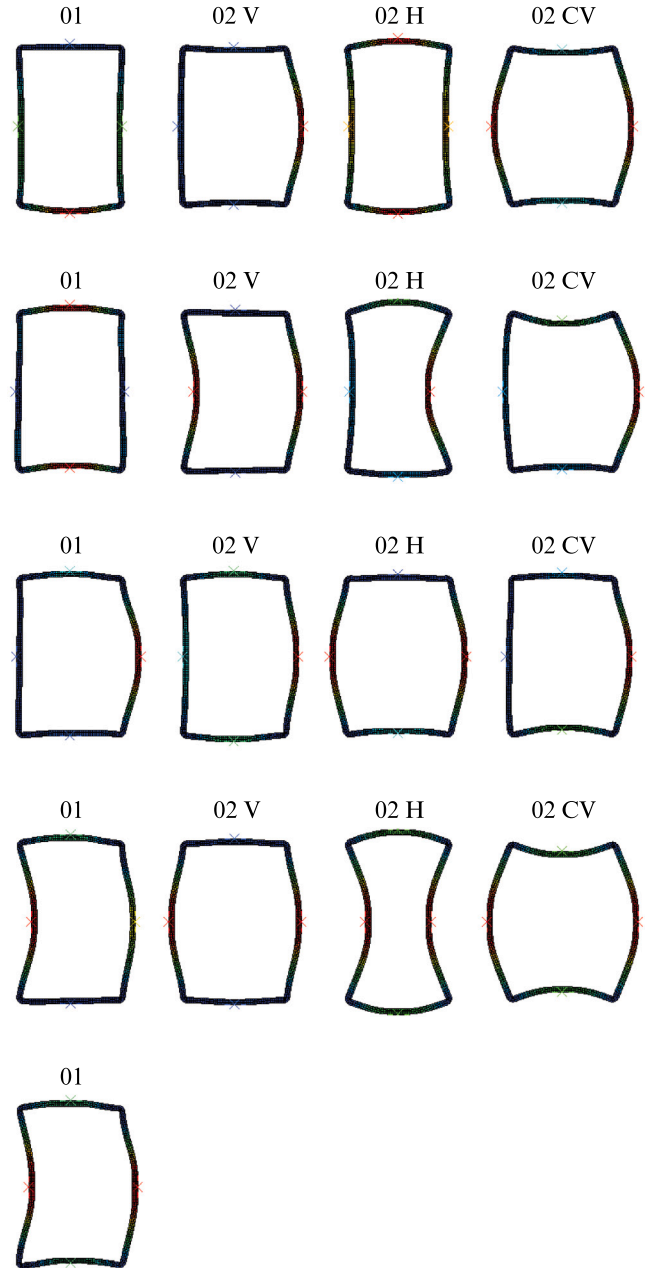


Fig. C.4. Cross-section deformation in W0 for RHS for all LPs with SC02.

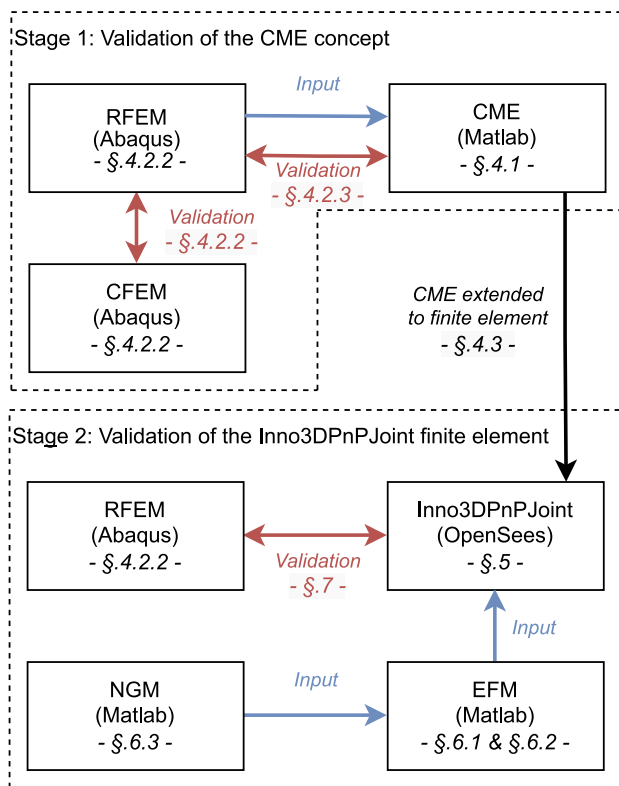


Fig. D.1. Workflow: mapping between analytical and finite element models.

References

- [1] European Commission, INNOvative 3D JOINTS for robust and economic hybrid tubular construction. Project carried out with a financial grant of the Research Programme of the Research Fund for Coal and Steel. Grant Agreement Number 749959, Research Programme of the Research Fund for Coal and Steel, 2017-2021.
- [2] L. Simões da Silva, T. Tankova, H.D. Craveiro, L.C. Silva, R. Simões, R. Costa, Innovative 3D joint for steel modular construction, *Ce/Papers 4* (2–4) (2021) 958–963, <http://dx.doi.org/10.1002/cepa.1384>, URL <https://onlinelibrary.wiley.com/doi/10.1002/cepa.1384>.
- [3] L. Simões da Silva, L.C. Silva, T. Tankova, H.D. Craveiro, R. Simões, R. Costa, M. D'Aniello, R. Landolfo, Performance of modular hybrid cold-formed/tubular structural system, *Structures 30* (August 2020) (2021) 1006–1019, <http://dx.doi.org/10.1016/j.istruc.2021.01.066>.
- [4] T. Tankova, H. Craveiro, L. Simões da Silva, F. Ribeiro, R. Simões, C. Martins, R. Costa, L. Simões da Silva, Behaviour of plug-and-play joints between RHS columns and CFS trusses, *Structures 41* (2022) 1719–1745, <http://dx.doi.org/10.1016/j.istruc.2022.05.099>.
- [5] EN 1993-1-8:2005, Eurocode 3: Design of steel structures – Part 1-8: Design of joints, European Committee for Standardization, Brussels, 2005.
- [6] F. McKenna, G. Fenves, M. Scott, Open System for Earthquake Engineering Simulation, University of California, Berkeley, 2000, URL <http://opensees.berkeley.edu>.
- [7] F. McKenna, M.H. Scott, G.L. Fenves, Nonlinear finite-element analysis software architecture using object composition, *J. Comput. Civ. Eng.* 24 (1) (2010) 95–107, [http://dx.doi.org/10.1061/\(ASCE\)CP.1943-5487.0000002](http://dx.doi.org/10.1061/(ASCE)CP.1943-5487.0000002).
- [8] J.-P. Jaspart, K. Weynand, Design of Joints in Steel and Composite Structures, Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim, Germany, 2016, <http://dx.doi.org/10.1002/9783433604762>.
- [9] L. Simões da Silva, A. Santiago, P. Vila Real, A component model for the behaviour of steel joints at elevated temperatures, *J. Constr. Steel Res.* 57 (11) (2001) 1169–1195, [http://dx.doi.org/10.1016/S0143-974X\(01\)00039-6](http://dx.doi.org/10.1016/S0143-974X(01)00039-6).
- [10] L. Simões da Silva, Towards a consistent design approach for steel joints under generalized loading, *J. Constr. Steel Res.* 64 (9) (2008) 1059–1075, <http://dx.doi.org/10.1016/j.jcsr.2008.02.017>.
- [11] M. Latour, V. Piluso, G. Rizzano, Cyclic modeling of bolted beam-to-column connections: Component approach, *J. Earthq. Eng.* 15 (4) (2011) 537–563, <http://dx.doi.org/10.1080/13632469.2010.513423>.
- [12] S. Oliveira, R. Costa, A. Shahbazian, C. Rebelo, Y. Harada, Component-based method for quasi-static cyclic behaviour of steel joints, *J. Constr. Steel Res.* 181 (2021) <http://dx.doi.org/10.1016/j.jcsr.2021.106551>.
- [13] L. Simões da Silva, S. Oliveira, R.J.T. Costa, F. Gentili, Design and analysis of steel structures considering the 3D behaviour of the joints, *Adv. Steel Construct.* 16 (2) (2020) 137–145, <http://dx.doi.org/10.18057/IJASC.2020.16.2.5>.
- [14] J.M. Castro, A.Y. Elghazouli, B.A. Izzuddin, Modelling of the panel zone in steel and composite moment frames, *Eng. Struct.* 27 (1) (2005) 129–144, <http://dx.doi.org/10.1016/j.engstruct.2004.09.008>.
- [15] J. Henriques, F. Gentili, L. Simões da Silva, R. Simões, Component based design model for composite beam to reinforced concrete wall moment-resistant joints, *Eng. Struct.* 87 (2015) 86–104, <http://dx.doi.org/10.1016/j.engstruct.2014.12.039>.
- [16] R. Costa, P. Providência, A. Dias, Component-based reinforced concrete beam-column joint model, *Struct. Concr.* 18 (2017) 164–176, <http://dx.doi.org/10.1002/suco.201600024>.
- [17] K. Weynand, J.-P. Jaspart, M. Steenhuis, The stiffness model of revised annex j of eurocode 3, 1996, <http://dx.doi.org/10.1016/b978-008042821-5/50100-0>.
- [18] E. Bayo, J. Gracia, B. Gil, R. Goni, An efficient cruciform element to model semirigid composite connections for frame analysis, *J. Constr. Steel Res.* 72 (2012) 97–104, <http://dx.doi.org/10.1016/j.jcsr.2011.11.006>.
- [19] E. Bayo, A. Loureiro, M. Lopez, Performance of cruciform finite elements that model 2D steel joints with beams of unequal depth in frame analysis, *Ce/Papers 1* (2–3) (2017) 729–738, <http://dx.doi.org/10.1002/cepa.112>.
- [20] E. Bayo, A. Loureiro, M. Lopez, L. Simões da Silva, General component based cruciform finite elements to model 2D steel joints with beams of equal and different depths, *Eng. Struct.* 152 (2017) 698–708, <http://dx.doi.org/10.1016/j.engstruct.2017.09.042>.
- [21] E. Bayo, J. Gracia, Stiffness modelling of 2D welded joints using metamodels based on mode shapes, *J. Constr. Steel Res.* 156 (2019) 242–251, <http://dx.doi.org/10.1016/j.jcsr.2019.02.017>.
- [22] E. Bayo, J. Gracia, Characterization of the behaviour of welded steel joints through modal components, *Ce/Papers 3* (3–4) (2019) 331–336, <http://dx.doi.org/10.1002/cepa.1062>.
- [23] R.J.T. Costa, F. Gentili, L. Simões da Silva, Simplified model for connections of steel structures in OpenSees, in: *Nordic Steel Construction Conference*, 2015, pp. 1–10, January.
- [24] Y. Harada, L. Simões da Silva, Approach to macro-modeling of 3D tubular column-to-beam joint: An extension of component method for joint modelling, in: *Connections in Steel Structures VIII*, American Institute of Steel Construction, Boston, Massachusetts, 2016, pp. 163–172.
- [25] Y. Harada, L. Simões da Silva, Three-dimensional macro-modeling of beam-to-rectangular hollow section column joints under cyclic loading. Part 1: Modeling of cyclic out-of-plane behavior of single isolated plate element, *J. Constr. Steel Res.* 162 (2019) 105713, <http://dx.doi.org/10.1016/j.jcsr.2019.105713>.
- [26] Y. Harada, L. Simões da Silva, Three-dimensional macro-modeling of beam-to-rectangular hollow section column joints under cyclic loading. Part 2: Modeling of beam-to-column joint by extended component-based approach, *J. Constr. Steel Res.* 162 (2019) <http://dx.doi.org/10.1016/j.jcsr.2019.105714>.
- [27] GitHub, Opensees: Joint element library, 2022, URL <https://github.com/OpenSees/OpenSees/tree/master/SRC/element/joint>.
- [28] L.N. Lowes, N. Mitra, A. Altoontash, A Beam-Column Joint Model for Simulating the Earthquake Response of Reinforced Concrete Frames, Technical Report, Pacific Earthquake Engineering Research Center University of California, Berkeley, 2003, p. 69, August.
- [29] L.N. Lowes, A. Altoontash, Modeling reinforced-concrete beam-column joints subjected to cyclic loading, *J. Struct. Eng.* 129 (12) (2003) 1686–1697, [http://dx.doi.org/10.1061/\(ASCE\)0733-9445\(2003\)129:12\(1686\)](http://dx.doi.org/10.1061/(ASCE)0733-9445(2003)129:12(1686)).
- [30] A. Altoontash, Simulation and damage models for performance assessment of reinforced concrete beam-column joints (Ph.D. thesis), Stanford University, 2004, February.
- [31] C.-Y. Seo, Y.-C. Lin, R. Sause, J. Ricles, Development of analytical models for 0.6 scale self-centering MRF with beam web friction devices, in: *6th International Conference for Steel Structures in Seismic Area (STESSA)*, CRC Press, Philadelphia, 2009, pp. 849–854.
- [32] T.L. Karavasilis, J. Ricles, C.-Y. Seo, Hybridfem : A program for dynamic time history analysis of 2D inelastic framed structures and real-time hybrid simulation., 2010.
- [33] P. Alanjari, B. Asgarian, N. Salari, Elastic tubular joint element for modelling of multi-brace, uni-planar tubular connections, *Ships Offshore Struct.* 10 (4) (2015) 404–415, <http://dx.doi.org/10.1080/17445302.2014.942077>.
- [34] Hibbit, Karlsson, Sorenson, ABAQUS/CAE user's manual, 2000.
- [35] Computers, Structures Incorporated (CSI), CSI analysis reference manual for SAP2000®, etabs®, safe® and csbridge, 2020.
- [36] Autodesk, Robot structural analysis professional 2021 user's guide, 2020.
- [37] G.M. Barsan, C.G. Chiorean, Computer program for large deflection elasto-plastic analysis of semi-rigid steel frameworks, *Comput. Struct.* 72 (6) (1999) 699–711, [http://dx.doi.org/10.1016/S0045-7949\(98\)00310-1](http://dx.doi.org/10.1016/S0045-7949(98)00310-1).

- [38] European Commission, INNOvative 3D JOINTS for Robust and Economic Hybrid Tubular Construction. Deliverable D4.1 Report on Components Assembly and on the Model Leading To the Construction of the Joint Macro Finite Element, Research Programme of the Research Fund for Coal and Steel, 2021.
- [39] European Commission, INNOvative 3D JOINTS for Robust and Economic Hybrid Tubular Construction. Deliverable D7.1 Report on New Rules for Hybrid Joints Based in the Component Method, Research Programme of the Research Fund for Coal and Steel, 2021.
- [40] T.M. Nguyen, A. Rodier, Macro-modelling of 3D tubular column-to-truss beam joints, *Ce/Papers* 4 (2–4) (2021) 924–928.
- [41] C.V. Miculaş, R.J. Costa, L. Simões da Silva, R. Simões, H. Craveiro, T. Tankova, Macro-modelling of the three-dimensional interaction between the faces of a steel tubular column joint, in: F. Di Trapani, C. Demartino, G.C. Marano, G. Monti (Eds.), *Proceedings of the 2022 Eurasian OpenSees Days*, Springer Nature Switzerland, Cham, 2023, pp. 408–422, http://dx.doi.org/10.1007/978-3-031-30125-4_37.
- [42] C.V. Miculaş, abqTCphd: an Abaqus RSG plug-in for effortless creation of tubular columns with socket faces, 2023, <http://dx.doi.org/10.5281/zenodo.7867301>.
- [43] G. Van Rossum, F.L. Drake, et al., *Python Reference Manual*, Centrum voor Wiskunde en Informatica Amsterdam, 1995.
- [44] C.V. Miculaş, abqTCphd: an Abaqus Plug-in to "Create Tubular Columns with Socket Faces" #1 download & installation, 2023, URL <https://youtu.be/LkeMnk1fEeo>.
- [45] C.V. Miculaş, abqTCphd: an Abaqus Plug-in to "Create Tubular Columns with Socket Faces" #2 model generation, 2023, URL <https://youtu.be/XaNEqAroR1c>.
- [46] T. Tankova, H. Craveiro, L.C. Silva, F.F. Ribeiro, R. Simões, C. Martins, R. Costa, L. Simões da Silva, Behaviour of plug-and-play joints between RHS columns and cfs trusses, *Structures* 41 (May) (2022) 1719–1745, <http://dx.doi.org/10.1016/j.istruc.2022.05.099>.
- [47] P. Krolo, D. Grandić, M. Bulić, The guidelines for modelling the preloading bolts in the structural connection using finite element methods, *J. Comput. Eng.* 2016 (2016) 1–8, <http://dx.doi.org/10.1155/2016/4724312>.
- [48] C.V. Miculaş, Innovative plug and play joints for hybrid tubular constructions (Ph.D. thesis), University of Coimbra, Portugal, 2023, URL <https://estudogeral.uc.pt/handle/10316/110990>.
- [49] T. Togo, Experimental Study on Mechanical Behavior of Tubular Joints (in Japanese) (Ph.D. thesis), Osaka University, 1967.
- [50] L. Costa Neves, F. Gomes, Semi-rigid behaviour of beam-to-column minor- axis joints, in: *IABSE Int.Colloquium on Semi-Rigid Structural Connections*, 1996, pp. 207–216.
- [51] L. Simões da Silva, L.F.N. Neves, F.C.T. Gomes, Rotational stiffness of rectangular hollow sections composite joints, *J. Struct. Eng.* 129 (4) (2003) 487–494, [http://dx.doi.org/10.1061/\(ASCE\)0733-9445\(2003\)129:4\(487\)](http://dx.doi.org/10.1061/(ASCE)0733-9445(2003)129:4(487)).
- [52] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, P. Ivanov, D. Avila, S. Abdalla, C. Willing, Jupyter notebooks – a publishing format for reproducible computational workflows, in: F. Loizides, B. Schmidt (Eds.), *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, IOS Press, 2016, pp. 87–90.
- [53] C.V. Miculaş, Tube Components Initial Stiffness Calculator for Inno3DPnJoint Element from OpenSees, 2023, <http://dx.doi.org/10.5281/zenodo.7869445>.
- [54] Project Jupyter, Matthias Bussonnier, Jessica Forde, Jeremy Freeman, Brian Granger, Tim Head, Chris Holdgraf, Kyle Kelley, Gladys Nalvarte, Andrew Osheroff, M. Pacer, Yuvi Panda, Fernando Perez, Benjamin Ragan-Kelley, Carol Willing, Binder 2.0 - reproducible, interactive, sharable environments for science at scale, in: Fatih Akici, David Lippa, Dillon Niederhut, M. Pacer (Eds.), *Proceedings of the 17th Python in Science Conference*, 2018, pp. 113–120, <http://dx.doi.org/10.25080/Majora-4a1f417-011>.
- [55] C.V. Miculaş, Inno3DPnJoint element: Tube components initial stiffness calculator with jupyter notebook & binder, 2023, URL <https://youtu.be/bsL44rdOQvc>.
- [56] B. Stroustrup, *The C++ Programming Language*, third ed., Addison-Wesley Longman Publishing Co., Inc., USA, 2000.
- [57] M. Corporation, *Visual studio* 2019, 2019.