FORMULATION & IMPLEMENTATION OF THREE DIMENSIONAL DOUBLY SYMMETRIC BEAM-COLUMN ANALYSES WITH WARPING EFFECTS IN OPENSEES

XI ZHANG
KIM JR RASMUSSEN
HAO ZHANG

RESEARCH REPORT R917
JULY 2011

ISSN 1833-2781

SCHOOL OF CIVIL ENGINEERING

THE UNIVERSITY OF SYDNEY

THE UNIVERSITY OF SYDNEY

SCHOOL OF CIVIL ENGINEERING

FORMULATION AND IMPLEMENTATION OF THREE-DIMENSIONAL DOUBLY SYMMETRIC BEAM-COLUMN ANALYSES WITH WARPING EFFECTS IN OPENSEES

RESEARCH REPORT R917

XI ZHANG
KIM J. R. RASMUSSEN
HAO ZHANG

July 2011

ISSN 1833-2781

School of Civil Engineering, Research Report R???
Title of Research Report here
Names of authors here
April 2006

ISSN 1833-2781

Published by:
School of Civil Engineering
The University of Sydney
Sydney NSW 2006
Australia

This report and other Research Reports published by the School of Civil Engineering are available at http://sydney.edu.au/civil

ABSTRACT

OpenSees is an object-oriented framework for finite element analysis. A key feature of OpenSees is the ability to integrate existing libraries and new components into the framework without the need to change the existing code. The non-linear beam-column element theory in OpenSees is based on the assumption that torsion is uniform through the length of the member and non-uniform warping torsion is excluded. In this report, a co-rotational mapping for three-dimensional analyses of doubly symmetric beam-column elements with warping effects is incorporated into OpenSees. Both a local linear strain assumption and a second order approximation of the Green-Lagrange strains are considered in the formulations of the beam elements at local level. Numerical examples are presented to demonstrate the accuracy of the proposed elements by comparisons with results from the literature and commercial softwares.

KEYWORDS

Co-rotational formulation, Warping, OpenSees, Thin-walled structures

TABLE OF CONTENTS

INTRODUCTION

In one-dimensional beam-column element approaches for modelling large displacement and finite rotation problems, there are typically three main approaches: Total Lagrangian (TL) formulation [1-4], updated Lagrangian (UL) formulation [5] and co-rotational Lagrangian (CL) formulation [6-12]. The co-rotational Lagrangian (CL) technique, first presented by Oran [13, 14], is to decompose the motion of the element into rigid body movements and pure deformations, through the use of the current deformed configuration as the reference system, which continuously rotates and translates with the element [9]. The deformational response is captured at the level of the local reference frame, whereas the geometric non-linearity induced by large displacements is considered in the transformation matrices relating local and global quantities. One of the main features of the co-rotational Lagrangian (CL) formulation is its independence of the local system while deriving the internal forces and stiffness matrices. It means that for elements with the same number of nodes and degrees of freedom the mapping from the local to the global system is the same. Thus the co-rotational Lagrangian (CL) formulation is considered as much simpler in concept compared with the total Lagrangian (TL) formulation and the updated Lagrangian (UL) formulation.

The Open System for Earthquake Engineering System (OpenSees) [15] is an object-oriented, open source software package which is widely used by researchers throughout the world. Its initial application aim is to simulate the response of structural systems subjected to earthquakes, but the software is also applicable to a number of non-seismic analyses. Being open source the software is modified and updated easily by researchers, and encourages free communication and interaction between its developers and end-users.

The co-rotational Lagrangian (CL) formulation technique is adopted in OpenSees for formulating non-linear beam-column elements. The beam-column elements in OpenSees include Elastic Beam-Column Element, Force Based Beam-Column Element, Beam with Hinges Element, Displacement Based Beam-Column Element and others.

The Elastic Beam-Column Element assumes elastic material throughout the analysis, and the linear strain assumption is used in formulating the element at local level. The Displacement Based Beam-Column Element utilises a displacement-based formulation in which cubic functions are used, for the interpolation of transverse displacements and the angle of twist while a linear function is used for the axial displacement. The linear strain assumption is utilised in terms of the local element formulation.

In the present report, both the code for Elastic Beam-Column Element and Displacement Based Beam-Column Element are modified and reincorporated to take account of warping effects. The new elements are called Elastic Beam-Column Warping Element and Displacement Based Beam-Column Warping Element, respectively. The additional warping degrees of freedom are introduced to the local elements. Since the warping is in itself a deformational quantity, these additional degrees of freedom remain constant during the sequence of transformations from the local to the global system, which means the warping degrees of freedom only have an effect on the local element formulation. The formulation of local element quantities and the mapping between local elements and global elements are based on the work of Crisfield [11, 16] and Alemdar [17].

CO-ROTATIONAL THEORY FOR LARGE DISPLACEMENT ANALYSIS OF DOUBLY SYMMETRIC BEAM-COLUMNS WITH WARPING EFFECTS

A CO-ROTATIONAL MAPPING FOR THREE-DIMENSIONAL LARGE DISPLACEMENT ANALYSIS

In three-dimensional large displacement analysis, the handling of spatial finite rotations is challenging and not a simple extension of two-dimensional analysis. This is because finite rotations in three-dimensional analysis are not true vector quantities, i.e. they do not comply with the rules of vector operations and the result will depend on the order in which the rotations are taken [6].

Basic theories involving finite rotations

Before introducing the element formulation, a few basic equations related to finite and large rotation will be presented. If no specific reference is given, the detailed explanation can be found in [11, 18]. The key concept associated with finite rotations involves an orthogonal transformation that rotates a position vector $\mathbf{r}_0$ into $\mathbf{r}_n$ through a 'pseudo-vector' $\boldsymbol{\theta}$ (Figure 1).

$$\mathbf{r}_n = \mathbf{R}(\boldsymbol{\theta})\mathbf{r}_0 \tag{1}$$

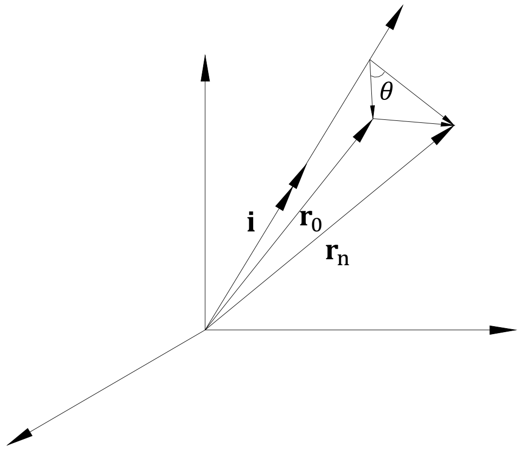

Figure 1: Finite rotation of a vector

The rotation matrix $\mathbf{R}(\boldsymbol{\theta})$ can be expressed as [16]

$$\mathbf{R}(\boldsymbol{\theta}) = \mathbf{I} + \frac{sin\theta}{\theta}\mathbf{S}(\boldsymbol{\theta}) + \frac{1 - cos\theta}{\theta^2}\mathbf{S}(\boldsymbol{\theta})\mathbf{S}(\boldsymbol{\theta}) \tag{2}$$

where the 'pseudo-vector' $\boldsymbol{\theta}$ is defined as $\boldsymbol{\theta} = \theta\mathbf{i}$ in which θ is the magnitude of the rotation about an axis directed along the unit vector $\mathbf{i}$, i.e.

$$\boldsymbol{\theta} = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{pmatrix} = \theta_1\mathbf{i}_1 + \theta_2\mathbf{i}_2 + \theta_3\mathbf{i}_3 = \theta\mathbf{i} \tag{3}$$

The term $\mathbf{S}(\boldsymbol{\theta})$ is called the skew-symmetric matrix and is defined as

$$\mathbf{S}(\boldsymbol{\theta}) = \begin{bmatrix} 0 & -\theta_3 & \theta_2 \\ \theta_3 & 0 & -\theta_1 \\ -\theta_2 & \theta_1 & 0 \end{bmatrix} \tag{4}$$

For compound rotations, because the rotations are non-additive, several special techniques are introduced. If $\mathbf{r}_1 = \mathbf{R}_1(\boldsymbol{\theta}_1)\mathbf{r}_0$ is followed by $\mathbf{r}_2 = \mathbf{R}_2(\boldsymbol{\theta}_2)\mathbf{r}_1$, we have

$$\mathbf{r}_2 = \mathbf{R}_2(\boldsymbol{\theta}_2)\mathbf{R}_1(\boldsymbol{\theta}_1)\mathbf{r}_1 = \mathbf{R}_{12}(\boldsymbol{\theta}_{12}) \tag{5}$$

As recommended by Crisfield [11], in order to compute $\mathbf{R}_{12}$, it is useful to adopt normalized quartonians, so that

$$\widehat{\boldsymbol{q}} = \begin{bmatrix} \mathbf{q} \\ \bar{q} \end{bmatrix} = \begin{bmatrix} \sin\left(\frac{\theta}{2}\right)\mathbf{i} \\ \cos\left(\frac{\theta}{2}\right) \end{bmatrix} \tag{6}$$

The quartonian can then be updated via

$$\hat{\mathbf{q}}_{12} = \begin{bmatrix} \mathbf{q}_{12} \\ \bar{q}_{12} \end{bmatrix} = \begin{bmatrix} \bar{q}_1 \mathbf{q}_2 + \bar{q}_2 \mathbf{q}_1 - \mathbf{q}_1 \times \mathbf{q}_2 \\ \bar{q}_1 \bar{q}_2 - \mathbf{q}_1^{\mathrm{T}} \mathbf{q}_2 \end{bmatrix} \tag{7}$$

The rotation matrix can be expressed in terms of the quartonian via

$$\mathbf{R} = (\bar{q}^2 - \mathbf{q}^{\mathrm{T}}\mathbf{q})\mathbf{I} + 2\mathbf{q}\mathbf{q}^{\mathrm{T}} + 2\bar{q}\mathbf{S}(\mathbf{q}) \tag{8}$$

The equations listed above are some of the basic equations used for calculating three-dimensional finite rotations. The equations in section 'Computing the local displacement' and section 'Computing the transformation matrix connecting local and global variables' are mostly derived from them. However, as they are not a key part of this report, the detailed derivation is not listed.

Global and local systems

The proposed theory has seven global degrees of freedom at each end of the element: three displacements, three rotations and one warping degree of freedom. Because there are two nodes in each element, the total degrees of freedom are fourteen. In the local element, rigid body modes are removed and only considered in the transformation from the local to the global system. Therefore there are only nine local degrees of freedom: the axial elongation (e), six local rotations relative to the element chord at each end ($\theta_{l1}, \theta_{l2}, \theta_{l3}, \theta_{l4}, \theta_{l5}, \theta_{l6}$) and two warping torsions at each end ($\theta'_{b1}, \theta'_{b2}$), as shown in Figure 2. The element local rotations are the element local slopes referred to the axis which connects the two ends of the element. The global rotations are spatial rotations or non-additive finite rotations referring to the global coordinate of system. The vector of global degrees of freedom $\mathbf{N}$ and the vector of local degrees of freedom $\mathbf{n}$ are defined as follows

$$\mathbf{N} = \{u_1, v_1, w_1, \omega_1, \omega_2, \omega_3, \theta'_{b1}, u_2, v_2, w_2, \omega_4, \omega_5, \omega_6, \theta'_{b2}\}^T \tag{9}$$

$$\mathbf{n} = \{\theta_{l1}, \theta_{l2}, \theta_{l3}, \theta'_{b1}, \theta_{l4}, \theta_{l5}, \theta_{l6}, \theta'_{b2}, e\}^T \tag{10}$$

The corresponding work conjugate forces and moments are

$$\mathbf{P} = \{F_1, F_2, F_3, M_1, M_2, M_3, M_{b1}, F_4, F_5, F_6, M_4, M_5, M_6, M_{b2}\}^T \tag{11}$$

$$\mathbf{p} = \{M_{l1}, M_{l2}, M_{l3}, M_{b1}, M_{l4}, M_{l5}, M_{l6}, M_{b2}, N_l\}^T \tag{12}$$

where the subscript 'l' denotes the variable refers to the local system.

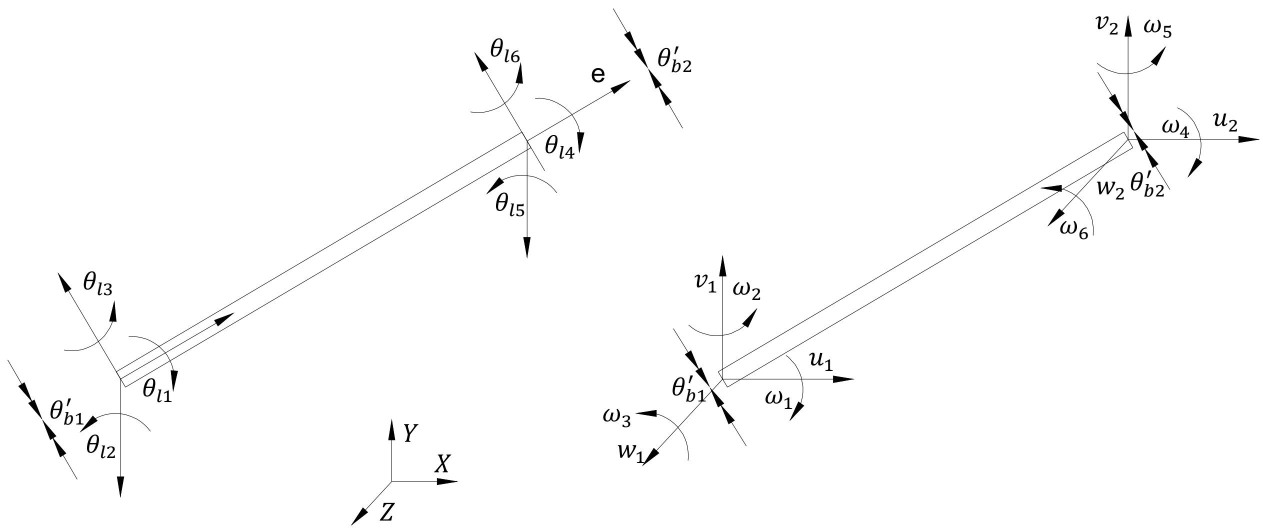

Figure 2 Element local degrees of freedom and global degrees of freedom

Computing the local displacements

In the co-rotational formulation, the rigid body motions are removed and the global rotations are transformed to the local system. Three triads are defined, two node triads at the end nodes of the element $\mathbf{T} = [\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3]$,

$\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3]$, and one local element base triad $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3]$, as shown in **Error! Reference source not found.**.

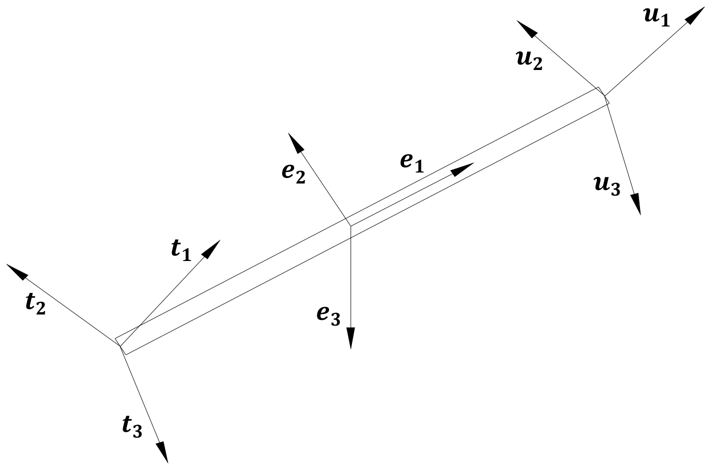

Figure 3 Three node triads of the element

The matrix $\mathbf{T}$ can be constructed with the help of Eq. (2) by using three updated global rotations (θ_1, θ_2, θ_3) of node 1 and $\mathbf{U}$ by using (θ_4, θ_5, θ_6) of node 2. Readers may refer to Crisfield [16] for detailed information. In the current configuration, the element base vector $\mathbf{e_1}$ is given by

$$\mathbf{e}_1 = \frac{1}{L}(\mathbf{X}_{21} + \mathbf{d}_{21}) \tag{13}$$

where L is the current length of the element, $\mathbf{X}_{21} = \mathbf{X}_2 - \mathbf{X}_1$ contains the initial coordinates of the element and $\mathbf{d}_{21} = \mathbf{d}_2 - \mathbf{d}_1$ is the element end displacement vector. The element bases $\mathbf{e}_2$ and $\mathbf{e}_3$, which give the orientation of the element cross-section after deformation, are defined as

$$\mathbf{e}_2 = \mathbf{r}_2 - \frac{\mathbf{r_2}^{\mathrm{T}}\mathbf{e}_1}{2}(\mathbf{e}_1 + \mathbf{r}_1) \tag{14}$$

$$\mathbf{e}_3 = \mathbf{r}_3 - \frac{\mathbf{r}_3{}^{\mathbf{T}}\mathbf{e}_1}{2}(\mathbf{e}_1 + \mathbf{r}_1) \tag{15}$$

where $\overline{\mathbf{R}} = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3]$ is the mean rotation matrix. Readers can refer to Crisfield [16] for more details for calculating $\overline{\mathbf{R}}$.

Once the three triads are achieved, the “local rotations” can be determined as follows,

$$\begin{aligned}
2\sin\theta_{l1} &= -\mathbf{t}_3^{\mathrm{T}}\mathbf{e}_2 + \mathbf{t}_2^{\mathrm{T}}\mathbf{e}_3 \\
2\sin\theta_{l2} &= -\,\mathbf{t}_2^{\mathrm{T}}\mathbf{e}_1 + \mathbf{e}_2^{\mathrm{T}}\mathbf{t}_1 \\
2\sin\theta_{l3} &= -\,\mathbf{t}_3^{\mathrm{T}}\mathbf{e}_1 + \mathbf{e}_3^{\mathrm{T}}\mathbf{t}_1 \\
2\sin\theta_{l4} &= -\,\mathbf{u}_3^{\mathrm{T}}\mathbf{e}_2 + \mathbf{u}_2^{\mathrm{T}}\mathbf{e}_3 \\
2\sin\theta_{l5} &= -\,\mathbf{u}_2^{\mathrm{T}}\mathbf{e}_1 + \mathbf{e}_2^{\mathrm{T}}\mathbf{u}_1 \\
2\sin\theta_{l6} &= -\,\mathbf{u}_3^{\mathrm{T}}\mathbf{e}_1 + \mathbf{e}_3^{\mathrm{T}}\mathbf{u}_1
\end{aligned} \tag{16}$$

The local axial displacement can be calculated from

$$e = L - L_0 = ((\mathbf{X}_{21} + \mathbf{d}_{21})^{\mathrm{T}}(\mathbf{X}_{21} + \mathbf{d}_{21}))^{1/2} - (\mathbf{X}_{21}^{\mathrm{T}}\mathbf{X}_{21})^{1/2} \tag{17}$$

where L_0 is the original length of the element.

Computing the transformation matrix connecting local and global variables

The relation between the variation of local and global variables can be defined as

$$\delta\mathbf{p} = \mathbf{F}\delta\mathbf{P} \tag{18}$$

where $\mathbf{F}$ is a 9×14 transformation matrix connecting the global and local systems. According to the work of Crisfield [16], the matrix $\mathbf{F}$ can be expressed as

$$\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3, \mathbf{f}_4, \mathbf{f}_5, \mathbf{f}_6, \mathbf{f}_7, \mathbf{f}_8, \mathbf{f}_9]^{\mathrm{T}} \tag{19}$$

where $\mathbf{f}_{\mathrm{i}}$ is ith row of the matrix $\mathbf{F}$. The $\mathbf{f}$ vectors associated with the local rotations are given by

$$\begin{aligned}
\mathbf{f}_1 &= \frac{1}{2\cos\theta_{l1}}(\mathbf{L}(\mathbf{r}_3)\mathbf{t}_2 - \mathbf{L}(\mathbf{r}_2)\mathbf{t}_3 + \mathbf{h}_1)\\
\mathbf{f}_2 &= \frac{1}{2\cos\theta_{l2}}(\mathbf{L}(\mathbf{r}_2)\mathbf{t}_1 + \mathbf{h}_2)\\
\mathbf{f}_3 &= \frac{1}{2\cos\theta_{l3}}(\mathbf{L}(\mathbf{r}_3)\mathbf{t}_1 + \mathbf{h}_3)\\
\mathbf{f}_4 &= [\mathbf{0}^{\mathrm{T}}, \mathbf{0}^{\mathrm{T}}, 1, \mathbf{0}^{\mathrm{T}}, \mathbf{0}^{\mathrm{T}}, 0]^{\mathrm{T}}\\
\mathbf{f}_5 &= \frac{1}{2\cos\theta_{l4}}(\mathbf{L}(\mathbf{r}_3)\mathbf{u}_2 - \mathbf{L}(\mathbf{r}_2)\mathbf{u}_3 + \mathbf{h}_4)\\
\mathbf{f}_6 &= \frac{1}{2\cos\theta_{l5}}(\mathbf{L}(\mathbf{r}_2)\mathbf{u}_1 + \mathbf{h}_5)\\
\mathbf{f}_7 &= \frac{1}{2\cos\theta_{l6}}(\mathbf{L}(\mathbf{r}_3)\mathbf{u}_1 + \mathbf{h}_6)\\
\mathbf{f}_8 &= [\mathbf{0}^{\mathrm{T}}, \mathbf{0}^{\mathrm{T}}, 0, \mathbf{0}^{\mathrm{T}}, \mathbf{0}^{\mathrm{T}}, 1]^{\mathrm{T}}\\
\mathbf{f}_9 &= [-\mathbf{e}_1^{\mathrm{T}}, \mathbf{0}^{\mathrm{T}}, 0, \mathbf{e}_1^{\mathrm{T}}, \mathbf{0}^{\mathrm{T}}, 0]^{\mathrm{T}}
\end{aligned} \tag{20}$$

The $\mathbf{L}$ matrix is given by

$$\mathbf{L}^{\mathrm{T}} = [\mathbf{L}_1^{\mathrm{T}}, \mathbf{L}_2^{\mathrm{T}}, 0, -\mathbf{L}_1^{\mathrm{T}}, \mathbf{L}_2^{\mathrm{T}}, 0] \tag{21}$$

$$\mathbf{L}_1(\mathbf{r}_{\mathrm{i}}) = \frac{\mathbf{r}_{\mathrm{i}}^{\mathrm{T}}\mathbf{e}_1}{2}\mathbf{A} + \frac{1}{2}\mathbf{A}\mathbf{r}_{\mathrm{i}}(\mathbf{e}_1 + \mathbf{r}_1)^{\mathrm{T}} \tag{22}$$

$$\mathbf{L}_2(\mathbf{r}_{\mathrm{i}}) = \frac{\mathbf{S}(\mathbf{r}_{\mathrm{i}})}{2} - \frac{\mathbf{r}_{\mathrm{i}}^{\mathrm{T}}\mathbf{e}_1}{4}\mathbf{S}(\mathbf{r}_1) - \frac{1}{4}\mathbf{S}(\mathbf{r}_{\mathrm{i}})\mathbf{e}_1(\mathbf{e}_1 + \mathbf{r}_1)^{\mathrm{T}} \tag{23}$$

where $\mathbf{S}(\mathbf{r}_{\mathrm{i}})$ is the skew-symmetric matrix of vector $\mathbf{r}_{\mathrm{i}}$ as defined in Eq. (4), $\mathbf{A}$ is a 3×3 symmetric matrix which is defined as

$$\mathbf{A} = \frac{1}{L}[\mathbf{I} - \mathbf{e}_1\mathbf{e}_1^{\mathrm{T}}] \tag{24}$$

and $\mathbf{h}_{\mathrm{i}}$ is are 14×1 vectors defined as follows,

$$\begin{aligned}
\mathbf{h}_1^{\mathrm{T}} &= \{\mathbf{0}^{\mathrm{T}}, (-\mathbf{S}(\mathbf{t}_3)\mathbf{e}_2 + \mathbf{S}(\mathbf{t}_2)\mathbf{e}_3)^{\mathrm{T}}, 0, \mathbf{0}^{\mathrm{T}}, \mathbf{0}^{\mathrm{T}}, 0\}\\
\mathbf{h}_2^{\mathrm{T}} &= \{(\mathbf{A}\mathbf{t}_2)^{\mathrm{T}}, (-\mathbf{S}(\mathbf{t}_2)\mathbf{e}_1 + \mathbf{S}(\mathbf{t}_1)\mathbf{e}_2)^{\mathrm{T}}, 0, -(\mathbf{A}\mathbf{t}_2)^{\mathrm{T}}, \mathbf{0}^{\mathrm{T}}, 0\}
\end{aligned} \tag{25}$$

$$\mathbf{h}_3^{\mathrm{T}} = \{(\mathbf{At}_3)^{\mathrm{T}}, (-\mathbf{S}(\mathbf{t}_3)\mathbf{e}_1 + \mathbf{S}(\mathbf{t}_1)\mathbf{e}_3)^{\mathrm{T}}, 0 - (\mathbf{At}_3)^{\mathrm{T}}, \mathbf{0}^{\mathrm{T}}, 0\}$$

$$\mathbf{h}_4^{\mathrm{T}} = \{\mathbf{0}^{\mathrm{T}}, \mathbf{0}^{\mathrm{T}}, 0, \mathbf{0}^{\mathrm{T}}, (-\mathbf{S}(\mathbf{u}_3)\mathbf{e}_2 + \mathbf{S}(\mathbf{u}_2)\mathbf{e}_3)^{\mathrm{T}}, 0\}$$

$$\mathbf{h}_5^{\mathrm{T}} = \{(\mathbf{Au}_2)^{\mathrm{T}}, \mathbf{0}^{\mathrm{T}}, 0, -(\mathbf{Au}_2)^{\mathrm{T}}, (-\mathbf{S}(\mathbf{u}_2)\mathbf{e}_1 + \mathbf{S}(\mathbf{u}_1)\mathbf{e}_2)^{\mathrm{T}}, 0\}$$

$$\mathbf{h}_6^{\mathrm{T}} = \{(\mathbf{Au}_3)^{\mathrm{T}}, \mathbf{0}^{\mathrm{T}}, 0, -(\mathbf{Au}_3)^{\mathrm{T}}, (-\mathbf{S}(\mathbf{u}_3)\mathbf{e}_1 + \mathbf{S}(\mathbf{u}_1)\mathbf{e}_3)^{\mathrm{T}}, 0\}$$

Having computed the transformation matrix $\mathbf{F}$, the global internal force vector can be simply computed from $\mathbf{P} = \mathbf{F}^{\mathrm{T}}\mathbf{p}$.

Computing the global tangent stiffness matrix

The global tangent stiffness equation may be obtained by differentiation of the expression for the global internal force vector,

$$\delta\mathbf{P} = \delta(\mathbf{F}^{\mathrm{T}}\mathbf{p}) = \mathbf{F}^{\mathrm{T}}\delta\mathbf{p} + \delta\mathbf{F}^{\mathrm{T}}\mathbf{p} = \mathbf{F}^{\mathrm{T}}\mathbf{K}_l\mathbf{F}\delta\mathbf{N} + \mathbf{K}_{t\sigma}\delta\mathbf{N} = (\mathbf{K}_{t1} + \mathbf{K}_{t\sigma})\delta\mathbf{N} = \mathbf{K}_t\delta\mathbf{N} \tag{26}$$

where $\mathbf{K}_t$ is referred to as the global stiffness matrix. The first term of $\mathbf{K}_t$ involves the local element stiffness matrix $\mathbf{K}_l$ and is expressed as

$$\mathbf{K}_{t1} = \mathbf{F}^{\mathrm{T}}\mathbf{K}_l\mathbf{F} \tag{27}$$

Once the expressions for $\mathbf{K}_l$ and $\mathbf{F}$ are known, the calculation of $\mathbf{K}_{t1}$ is straightforward. The derivation of the local element stiffness matrix $\mathbf{K}_l$ will be introduced in the following section.

The second term of $\mathbf{K}_t$ is referred to as the geometric stiffness matrix and accounts for large displacement effects. The detailed derivation of $\mathbf{K}_t$ is described by Crisfield [16].

DERIVATION OF LOCAL ELEMENT STIFFNESS MATRIX AND INTERNAL FORCES

One of the main features of the co-rotational Lagrangian (CL) formulation is its independence of the local system while deriving the internal forces and stiffness matrices. It means that the mapping from the local to the global system is the same for elements with the same number of nodes and degrees of freedom, and therefore the theory describing the transformation between the global and local systems can be used for any type of local element.

In this report, an element which is based on local linear strain and assumes elastic material throughout the analysis as well as one based on a second order approximation of the Green-Lagrange strains are developed. Both elements are capable of considering warping effects. To be consistent with the names of the corresponding existing elements in OpenSees, the first is called Elastic Beam-column Warping Element and the second is called Displacement Based Beam-Column Warping Element. Throughout the report, the two elements assume the cross-section is doubly symmetric.

Elastic Beam-Column Warping Element

The Elastic Beam-column Warping Element assumes elastic material throughout the analysis, and uses the linear strain assumption.

The formulations of the local element stiffness matrix and internal forces are based on typical engineering theory and are straightforward. The local stiffness matrix $\mathbf{K}_l$ can be expressed as [17],

$$\mathbf{K}_l = \begin{bmatrix} \frac{12EC_w}{L^3}+\frac{6GJ}{5L} & 0 & 0 & \frac{GJ}{10}+\frac{6EC_w}{L^2} & -\frac{12EC_w}{L^3}-\frac{6GJ}{5L} & 0 & 0 & \frac{GJ}{10}+\frac{6EC_w}{L^2} & 0 \\ 0 & \frac{4EI_z}{L} & 0 & 0 & 0 & \frac{2EI_z}{L} & 0 & 0 & 0 \\ 0 & 0 & \frac{4EI_y}{L} & 0 & 0 & 0 & \frac{2EI_y}{L} & 0 & 0 \\ \frac{GJ}{10}+\frac{6EC_w}{L^2} & 0 & 0 & \frac{4EC_w}{L}+\frac{2}{15}GJL & -\frac{GJ}{10}-\frac{6EC_w}{L^2} & 0 & 0 & \frac{2EC_w}{L}-\frac{1}{30}GJL & 0 \\ -\frac{12EC_w}{L^3}-\frac{6GJ}{5L} & 0 & 0 & -\frac{GJ}{10}-\frac{6EC_w}{L^2} & \frac{12EC_w}{L^3}+\frac{6GJ}{5L} & 0 & 0 & -\frac{GJ}{10}-\frac{6EC_w}{L^2} & 0 \\ 0 & \frac{2EI_z}{L} & 0 & 0 & 0 & \frac{4EI_z}{L} & 0 & 0 & 0 \\ 0 & 0 & \frac{2EI_y}{L} & 0 & 0 & 0 & \frac{4EI_y}{L} & 0 & 0 \\ \frac{1}{10}GJ+\frac{6EC_w}{L^2} & 0 & 0 & \frac{2EC_w}{L}+\frac{1}{30}GJL & -\frac{GJ}{10}-\frac{6EC_w}{L^2} & 0 & 0 & \frac{4EC_w}{L}+\frac{2}{15}GJL & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{EA}{L} \end{bmatrix} \tag{28}$$

The internal forces in the local element can be calculated by multiplying each element of the stiffness matrix by corresponding local displacements

$$\begin{aligned} M_{l1} = &\left(\frac{12EC_w}{L^3}+\frac{6GJ}{5L}\right)\theta_{l1}+\left(\frac{GJ}{10}+\frac{6EC_w}{L^2}\right)\theta_{b1} \\ &+\left(-\frac{12EC_w}{L^3}-\frac{6GJ}{5L}\right)\theta_{l4}+\left(\frac{GJ}{10}+\frac{6EC_w}{L^2}\right)\theta_{b2} \end{aligned} \tag{29}$$

$$M_{l2} = \frac{4EI_z}{L}\theta_{l2}+\frac{2EI_z}{L}\theta_{l5} \tag{30}$$

$$M_{l3} = \frac{4EI_y}{L}\theta_{l3}+\frac{2EI_y}{L}\theta_{l6} \tag{31}$$

$$\begin{aligned} M_{b1} = &\left(\frac{GJ}{10}+\frac{6EC_w}{L^2}\right)\theta_{l1}+\left(\frac{4EC_w}{L}+\frac{2}{15}GJL\right)\theta_{b1} \\ &+\left(-\frac{GJ}{10}-\frac{6EC_w}{L^2}\right)\theta_{l4}+\left(\frac{2EC_w}{L}-\frac{1}{30}GJL\right)\theta_{b2} \end{aligned} \tag{32}$$

$$\begin{aligned} M_{l4} = &-\left(\frac{12EC_w}{L^3}+\frac{6GJ}{5L}\right)\theta_{l1}-\left(\frac{GJ}{10}+\frac{6EC_w}{L^2}\right)\theta_{b1} \\ &+\left(\frac{12EC_w}{L^3}+\frac{6GJ}{5L}\right)\theta_{l4}-\left(\frac{GJ}{10}+\frac{6EC_w}{L^2}\right)\theta_{b2} \end{aligned} \tag{33}$$

$$M_{l5} = \frac{2EI_z}{L}\theta_{l2}+\frac{4EI_z}{L}\theta_{l5} \tag{34}$$

$$M_{l6} = \frac{2EI_y}{L}\theta_{l3}+\frac{4EI_y}{L}\theta_{l6} \tag{35}$$

$$\begin{aligned} M_{b2} = &\left(\frac{GJ}{10}+\frac{6EC_w}{L^2}\right)\theta_{l1}+\left(\frac{2EC_w}{L}+\frac{1}{30}GJL\right)\theta_{b1} \\ &+\left(-\frac{1}{10}GJ-\frac{6EC_w}{L^2}\right)\theta_{l4}+\left(\frac{4EC_w}{L}+\frac{2}{15}GJL\right)\theta_{b2} \end{aligned} \tag{36}$$

Non-linear geometric effects are ignored at the local element level and only considered in the transformation between local and global systems. As shown later in the report, the Elastic Beam-Column Element can predict accurate results for most cases. However, it is also shown that the element may give inaccurate results for problems where torsional effects are significant [9].

Displacement Based Beam-Column Warping Element

A general accurate beam-column formulation requires that second order terms are included in the strain-displacement relationships to handle large displacements. It is realized that if the terms corresponding to rigid body motions are successfully removed from the formulation, and if one further assumes small strains (but large displacements and large rotations for the element motion), simpler forms of the strain measures can be successfully implemented in the co-rotational formulation. The formulation of the displacement based beam-column element is based on the work of Alemdar [17].

In a doubly-symmetric thin-walled section, the approximation of the Green-Lagrange finite strain measure which includes second-order terms can be expressed as

$$\varepsilon = u' + \frac{1}{2}(v')^2 + \frac{1}{2}(w')^2 + \varpi\phi'' - yv'' - zw'' + \frac{1}{2}(y^2 + z^2)(\phi')^2 - y\phi w'' + z\phi v'' \tag{37}$$

where u, v, w are the displacements of the shear centre of the cross-section, ϕ is the rotation about the shear centre, (the centroid and the shear centre coincide for doubly-symmetric section), x, y, z are the coordinates of an arbitrary point of the cross-section in the principal axis system, and $\varpi = \varpi(y, z)$ is the sectorial area of the arbitrary point.

In Eq. (37), u' is the axial strain due to elongation, yv'' and zw'' are terms of axial strain due to bending about the principal axes, $y\phi w''$ and $z\phi v''$ are strains due to coupling between bending and torsion, $\frac{1}{2}(v')^2$ and $\frac{1}{2}(w')^2$ are strains due to coupling between axial strain and bending, $\frac{1}{2}(y^2 + z^2)(\phi')^2$ is the Wagner term which defines coupling between axial strain and torsion and $\varpi\phi''$ is the effect of torsional warping on the axial strain.

The variation of ε is expressed as

$$\begin{aligned}\delta\varepsilon = \delta u' + v'\delta v' + w'\delta w' - y(\delta v'' + w''\delta\phi + \phi\delta w'') + \\ z(-\delta w'' + v''\delta\phi + \phi\delta v'') + (y^2 + z^2)\phi'\delta\phi' + \varpi\delta\phi''\end{aligned} \tag{38}$$

In current formulation for thin-walled sections, shear strains due to bending and warping torsion are neglected, and the shear strain due to uniform torsion is assumed to vary linearly through the thickness of component plates with zero mid-plane value [19]

$$\tau = -2r\kappa_z \tag{39}$$

where r is a coordinate perpendicular to the tangent of the mid-surface at an arbitrary point. κ_z is the twist

$$\kappa_z = \phi' + \frac{1}{2}(u''v' - u'v'') \tag{40}$$

By further assuming the twist due to bending is ignored, the shear strain due to uniform torsion can be approximated by

$$\tau = -2r\phi' \tag{41}$$

and its variation is

$$\delta\tau = -2r\delta\phi' \tag{42}$$

The variation of the strain components may be expressed in a matrix form

$$\delta\boldsymbol{\varepsilon} = \begin{Bmatrix}\delta\varepsilon \\ \delta\tau\end{Bmatrix} = \begin{bmatrix}1 & -y & -z & y^2+z^2 & \varpi & 0 \\ 0 & 0 & 0 & 0 & 0 & -2r\end{bmatrix} \begin{bmatrix}1 & v' & w' & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \phi & w'' & 0 & 0 \\ 0 & 0 & 0 & -\phi & 1 & -v'' & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \phi' & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0\end{bmatrix} \begin{bmatrix}\delta u' \\ \delta v' \\ \delta w' \\ \delta v'' \\ \delta w'' \\ \delta\phi \\ \delta\phi' \\ \delta\phi''\end{bmatrix} \tag{43}$$

or

$$\delta\boldsymbol{\varepsilon} = \mathbf{Y}\mathbf{N}_{\delta d1}\delta\mathbf{v} \tag{44}$$

where $\mathbf{Y}$ and $\mathbf{N}_{\delta d1}$ are functions of the coordinates (y, z, ϖ, r) and generalised displacements (v, w, ϕ), respectively.

Shape functions are introduced in this report to express the axial displacement (u), the transverse displacements (v, w) and the twist rotation (ϕ). A linear function is chosen for the axial displacement, while cubic Hermitian functions are chosen for the transverse displacements and twist rotation,

$$\begin{aligned} u &= \mathbf{N}_u{}^{\mathrm{T}}\mathbf{n} \\ v &= \mathbf{N}_v{}^{\mathrm{T}}\mathbf{n} \\ w &= \mathbf{N}_w{}^{\mathrm{T}}\mathbf{n} \\ \phi &= \mathbf{N}_\phi{}^{\mathrm{T}}\mathbf{n} \end{aligned} \tag{45}$$

where $\mathbf{n}$ is the element nodal displacement vector in the local system as described in Eq. (10), and the shape functions can be expressed in the following form

$$\begin{aligned} \mathbf{N}_u{}^{\mathrm{T}} &= [0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ N_{u1}] \\ \mathbf{N}_v{}^{\mathrm{T}} &= [0\ \ \ N_{v1}\ \ 0\ \ 0\ \ 0\ \ N_{v2}\ \ 0\ \ 0\ \ 0] \\ \mathbf{N}_w{}^{\mathrm{T}} &= [0\ \ 0\ \ N_{w1}\ \ 0\ \ 0\ \ 0\ \ N_{w2}\ \ 0\ \ 0] \\ \mathbf{N}_\phi{}^{\mathrm{T}} &= [\,N_{\phi1}\ \ 0\ \ 0\ \ N_{\phi2}\ \ N_{\phi3}\ \ 0\ \ 0\ \ N_{\phi4}\ \ 0] \end{aligned} \tag{46}$$

where

$$\begin{aligned} N_{u1} &= \frac{x}{L} \\ N_{v1} = N_{w1} = N_{\phi2} &= x - \frac{2x^2}{L} + \frac{x^3}{L^2} \\ N_{v2} = N_{w2} = N_{\phi4} &= -\frac{x^2}{L} + \frac{x^3}{L^2} \\ N_{\phi1} &= 1 - \frac{3x^2}{L^2} + \frac{2x^3}{L^3} \\ N_{\phi3} &= \frac{3x^2}{L^2} - \frac{2x^3}{L^3} \end{aligned} \tag{47}$$

With the aid of Eq. (45), $\delta\mathbf{v}$ in Eq. (44) can be expressed using element nodal displacement vector $\mathbf{n}$

$$\boldsymbol{\delta}\mathbf{v} = \mathbf{N}_{\delta d2}\delta\mathbf{n} \tag{48}$$

where

$$\mathbf{N}_{\delta d2} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & N'_{u1} \\ 0 & N'_{v1} & 0 & 0 & 0 & N'_{v2} & 0 & 0 & 0 \\ 0 & 0 & N'_{w1} & 0 & 0 & 0 & N'_{w2} & 0 & 0 \\ 0 & N''_{v1} & 0 & 0 & 0 & N''_{v2} & 0 & 0 & 0 \\ 0 & 0 & N''_{w1} & 0 & 0 & 0 & N''_{w2} & 0 & 0 \\ N_{\phi 1} & 0 & 0 & N_{\phi 2} & N_{\phi 3} & 0 & 0 & N_{\phi 4} & 0 \\ N'_{\phi 1} & 0 & 0 & N'_{\phi 2} & N'_{\phi 3} & 0 & 0 & N'_{\phi 4} & 0 \\ N''_{\phi 1} & 0 & 0 & N''_{\phi 2} & N''_{\phi 3} & 0 & 0 & N''_{\phi 4} & 0 \end{bmatrix} \tag{49}$$

Therefore Eq. (44) becomes

$$\delta \boldsymbol{\varepsilon} = \mathbf{Y}\mathbf{N}_{\delta d1}\mathbf{N}_{\delta d2}\delta\mathbf{n} \tag{50}$$

The virtual work principle may be used in deriving local element stiffness matrix

$$\int_{V_0} \delta\boldsymbol{\varepsilon}^{\mathrm{T}}\boldsymbol{\sigma}\, dV = \delta\mathbf{N}^{\mathrm{T}}\mathbf{Q}_{ext} = \delta\mathbf{n}^{\mathrm{T}}\mathbf{F}\mathbf{Q}_{ext} \tag{51}$$

where $\mathbf{Q}_{ext}$ is the global external force vector. After substituting Eq. (50) into the above equation and rearranging, we have

$$\delta\mathbf{n}^{\mathrm{T}}\left\{\int_0^L (\mathbf{N}_{\delta d2}^{\mathrm{T}}\,\mathbf{N}_{\delta d1}^{\mathrm{T}}\mathbf{D})dx - \mathbf{F}\mathbf{Q}_{ext}\right\} = 0 \tag{52}$$

Therefore the internal force in the local system $\mathbf{p}$ can be calculated via

$$\mathbf{p} = \int_{\mathbf{0}}^{L} (\mathbf{N}_{\delta d2}^{\mathrm{T}}\,\mathbf{N}_{\delta d1}^{\mathrm{T}}\mathbf{D})dx \tag{53}$$

where $\mathbf{D} = [P\ \ M_z\ \ M_y\ \ W\ \ B\ \ T]^T$ is the section stress resultant vector, which can be computed from,

$$\mathbf{D} = \int_{A_0} \mathbf{Y}^{\mathrm{T}}\,\boldsymbol{\sigma}\, dA \tag{54}$$

and $\boldsymbol{\sigma}^{\mathrm{T}} = [\sigma, \tau]$ is the section stress vector, the variation of which is related to the variation of the local node displacement vector via

$$\delta\boldsymbol{\sigma} = \mathbf{E}\delta\boldsymbol{\varepsilon} = \mathbf{E}\mathbf{Y}\mathbf{N}_{\delta d1}\mathbf{N}_{\delta d2}\delta\mathbf{n} \tag{55}$$

The variation of Eq. (53) leads to the local stiffness matrix

$$\begin{aligned} \delta\mathbf{p} &= \delta\left\{\int_0^L (\mathbf{N}_{\delta d2}^{\mathrm{T}}\,\mathbf{N}_{\delta d1}^{\mathrm{T}}\mathbf{D})dx\right\} \\ &= \int_0^L (\mathbf{N}_{\delta d2}^{\mathrm{T}}\,\delta\mathbf{N}_{\delta d1}^{\mathrm{T}}\mathbf{D})dx + \int_{\mathbf{0}}^L (\mathbf{N}_{\delta d2}^{\mathrm{T}}\,\mathbf{N}_{\delta d1}^{\mathrm{T}}\delta\mathbf{D})dx \\ &= \left\{\int_0^L (\mathbf{N}_{\delta d2}^{\mathrm{T}}\,\mathbf{G}\mathbf{N}_{\delta d2})dx + \int_0^L (\mathbf{N}_{\delta d2}^{\mathrm{T}}\,\delta\mathbf{N}_{\delta d1}^{\mathrm{T}}\mathbf{K}\mathbf{N}_{\delta d1}\mathbf{N}_{\delta d2})dx\right\}\delta\mathbf{n} = \mathbf{K}_l\delta\mathbf{n} \end{aligned} \tag{56}$$

where $\mathbf{K}_l = \int_0^L (\mathbf{N}_{\delta d2}^{\mathrm{T}}\,\mathbf{G}\mathbf{N}_{\delta d2})dx + \int_0^L (\mathbf{N}_{\delta d2}^{\mathrm{T}}\,\delta\mathbf{N}_{\delta d1}^{\mathrm{T}}\mathbf{K}\mathbf{N}_{\delta d1}\mathbf{N}_{\delta d2})dx$ is the element local stiffness matrix, in which

$$\mathbf{G} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & P & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & P & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & M_y & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & M_z & 0 & 0 \\ 0 & 0 & 0 & M_y & M_z & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & W & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{57}$$

$$\mathbf{K} = \int_{A_0} \mathbf{Y}^{\mathrm{T}} \mathbf{E} \mathbf{Y} dA \tag{58}$$

The expression for the section force $\mathbf{D}$ and section tangent stiffness $\mathbf{K}$ will be described later. The overall solution strategy for the non-linear displacement based beam-column element is summarised as follows:

a. Update the displacement vector $\mathbf{d}^{\mathrm{i}+1} = \mathbf{d}^{\mathrm{i}} + \delta\mathbf{d}^{\mathrm{i}}$.
b. Update the nodal triads **T** and **U** with the help of the 'mean node triad' as described by Crisfield [16].
c. Compute $\mathbf{e}_1$, $\mathbf{e}_2$, $\mathbf{e}_3$ using Eqs. (13) - (15).
d. Compute the local rotations θ_l^{i+1} from Eq. (16) and the local axial displacement e from Eq. (17). The warping rotation $(\theta'_{b1}, \theta'_{b2})$ can be achieved directly from global displacement vector (the local warping torsion is equal to the global warping torsion).
e. Compute the strain at arbitrary points using Eq. (37), and then update the stress via stress-strain relationship.
f. Compute section forces from Eq. (54) using updated stresses, and then calculate the element internal force vector from Eq. (53).
g. Calculate the element local tangent stiffness matrix from Eq. (56).
h. Compute the transformation matrix $\mathbf{F}$ from Eqs. (19) - (25).
i. Compute the global internal force vector using $\mathbf{P} = \mathbf{F}^{\mathrm{T}}\mathbf{p}$.
j. Compute global stiffness matrix from Eq. (26), and determine the $\delta\mathbf{d}^{\mathrm{i}}$.
k. Check residual force (displacement). If the norm of the residual is smaller than specified maximum one, set $\mathbf{d}^{\mathrm{i}+2} = \mathbf{d}^{\mathrm{i}+1} + \delta\mathbf{d}^{\mathrm{i}+1}$ and commence the next increment. Otherwise go back to step a, change $\delta\mathbf{d}$ and change the iteration.

INCORPORATION OF LARGE DISPLACEMENT ANALYSIS THEORY INTO OPENSEES

Existing beam-column elements in OpenSees ignore non-uniform warping torsion. As thin-walled open sections are widely used, it is desirable to include elements that incorporate warping effects into the program thus enabling the solution of problems in which warping torsion plays an important role for the overall behaviour.

OpenSees is primarily written and developed in objected-oriented language C++. It is designed in a modular fashion to support the finite element method with loose coupling between modules, which allows developers to modify or incorporate new modules without changing other modules. Developers do not need to know the details of the complete framework and can focus on making improvements in the areas of their own expertise [20].

In the following sections, all the symbols with superscript '*' represent expressions in the original OpenSees source code.

A BRIEF INTRODUCTION TO THE OPENSEES FRAMEWORK

The design of the OpenSees framework is based on the concept of objects. Objects represent the state and behaviour of components in a system being modelled computationally. Each object is capable of receiving messages, processing data and sending messages to other objects. The software design process involves specifying computational abstractions and defining objects and relationships between objects to represent the abstractions. For example, the Element is a key object which is responsible for maintaining the element's state and operating functions of states, such as computing the tangent stiffness matrix, computing the resistance force, etc.

In addition to objects, the software builds associations between objects. For example, to compute the resisting force, the element object interacts with other objects which contain the constitutive behaviour of the section. The fundamental objects and their relationships in OpenSees are shown in Figure 4.

The Domain object represents the current state of the entire finite element model, which changes when calculating and updating the Analysis object. The ModelBuilder object is responsible for creating the information needed for the finite element analysis, and adding it to the Domain object. The Recorder object records and reports information from the Domain object which is used for post-processing and visualization of the simulation results. These high level objects are all composed of more detailed lower level objects. For example, the Domain object is an aggregation of Node, Element, Constraint and Load objects (see Figure 5).

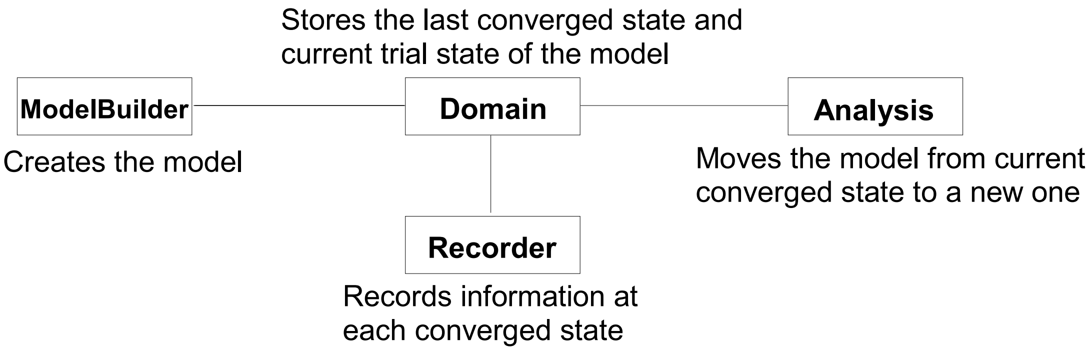

Figure 4 High level OpenSees objects, adapted from Fenves [20]

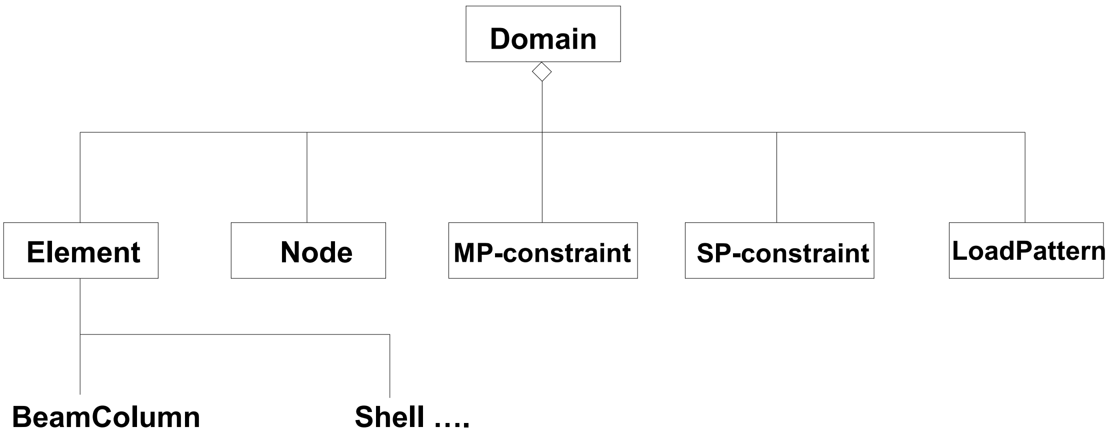

Figure 5 Components of the Domain object, adapted from Fenves [20]

MODIFYING THE FRAMEWORK TO ACCOMMODATE THE WARPING DEGREE OF FREEDOM

Most of users of OpenSees build models and conduct analyses via the string-based Tool Command Language (Tcl). Tcl is fully programmable with control structures, variable substitutions, and procedures that are necessary to automate routine operations using scripts. The general purpose of Tcl is to serve as a glue language that assembles software building blocks into customized applications. In the case of OpenSees, the Tcl interpreter is extended with commands to define the information of nodes, elements, boundary conditions, load and analysis algorithm, etc. The information defined in Tcl is then passed to the program to perform the analysis.

The Tcl script files are first modified to include information about the warping degree of freedom. The modifications include specifying 'ndf=7' (degrees of freedom for each node is 7), introducing warping constraints information to each node, adding information about the stress resultant associated with warping (bimoment), and if required, specifying the warping rigidity of the section (only for Elastic Beam-Column

Warping Element). An example of the modified Tcl file for Elastic Beam-Column Warping Element is shown in Appendix 1.
These modifications should be 'understood' by the OpenSees program, which is achieved by modifying the interpreter code of the OpenSees program. The interpreter code serves as the communication medium between the Tcl files and the analysis code. They are mostly in class TclModelBuilding. For example, the warping rigidity (C_w) is introduced in the Tcl script file lines for the Elastic Beam-Column Element to ensure the program 'understands' the meaning of C_w, i.e. the class TclElasticBeamCommand is modified as follows:

*theBeam = new ElasticBeamW3d (beamId,A,E,G,Jx,Iy,Iz,iNode,jNode,*theTrans,Cw,mass);*

Accordingly, the class definition of ElasticWBeam3d is also changed so that C_w can be used in the calculation of the local element.

The existing OpenSees code assumes six degrees of freedom for each node in three-dimensional analysis. In the process of building a model, the OpenSees source code first interacts with the Tcl script file to achieve information about the dimension, and if the dimension is three, OpenSees will automatically assign six degrees of freedom to each node. Besides, checks are performed so that an error is returned in case the number of degrees of freedom is other than six. For example, the following code is used to ensure the degrees of freedom of each node are six for three-dimensional analysis:

```
int dofNd1 = theNodes[0]->getNumberDOF();
 int dofNd2 = theNodes[1]->getNumberDOF();
 if (dofNd1 != 6 || dofNd2 != 6) {
 opserr << "FATAL ERROR DispBeamColumn3d (tag: %d), has differing number of DOFs at its  nodes"
```

This and similar code, which are mostly used in assigning node, element and boundary condition classes must be removed to relax this restriction.

MODIFYING THE MAPPING BETWEEN GLOBAL AND LOCAL SYSTEMS

In the original OpenSees source code, an abstract class GeometricTransformation is defined to provide multiple definitions of the displacement and force transformations between the global and local systems. Subclass CorotCrdTransf3d encapsulates nonlinear kinematic and equilibrium relationships that account for large displacements of the element [21].

In OpenSees, the global displacement vector $\mathbf{N}^*$ and local displacement vector $\boldsymbol{n}^*$ are defined as

$$\mathbf{N}^* = \{u_1, v_1, w_1, \omega_1, \omega_2, \omega_3, u_2, v_2, w_2, \omega_4, \omega_5, \omega_6\}^{\mathrm{T}} \tag{59}$$

$$\mathbf{n}^* = \{\theta_{l1}, \theta_{l2}, \theta_{l3}, \theta_{l4}, \theta_{l5}, \theta_{l6}, e\}^{\mathrm{T}} \tag{60}$$

The mapping between the global and local systems follows the theory presented above where the transformation matrix $\mathbf{F}^*$ is introduced to relate global and local values, i.e.

$$\mathbf{P}^* = \mathbf{F}^{*\mathrm{T}}\mathbf{p}^* \tag{61}$$

where $\mathbf{P}^*$, $\mathbf{p}^*$ and $\mathbf{F}^*$ are the global internal force, local internal force and transformation matrix, respectively, which do not account for warping effects.

The global stiffness matrix $\mathbf{K}_t^*$ can be achieved via

$$\mathbf{K}_t^* = \mathbf{F}^{*\mathrm{T}}\mathbf{K}_l^*\mathbf{F}^* + \mathbf{K}_{t\sigma}^* = \mathbf{K}_{t1}^* + \mathbf{K}_{t\sigma}^* \tag{62}$$

To introduce warping effects in the code, a new subclass CorotCrdTransfW3d is created, where $\mathbf{P}^*$, $\mathbf{p}^*$, $\mathbf{F}^*$ and $\mathbf{K}_{t\sigma}^*$ are replaced by $\mathbf{P}$, $\mathbf{p}$, $\mathbf{F}$ and $\mathbf{K}_{t\sigma}$, respectively, as shown in Eqs. (19) - (27).

Alternatively, as warping in itself is a deformational quantity, the warping degrees of freedom remain constant during the sequence of transformations. For clarification, if warping degrees of freedom are defined separately from non-warping degrees of freedom, $\mathbf{F}$ can be expressed as

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}^* & \mathbf{0}_2 \\ \mathbf{0}_2 & \mathbf{I}_2 \end{bmatrix} \tag{63}$$

where $\mathbf{I}_2$ donates a 2×2 identity matrix. Accordingly, $\mathbf{K}_{t\sigma}$ can be expressed as

$$\mathbf{K}_{t\sigma} = \begin{bmatrix} \mathbf{K}^*_{t\sigma} & \mathbf{0}_2 \\ \mathbf{0}_2 & \mathbf{0}_2 \end{bmatrix} \tag{64}$$

MODIFYING THE PLASTIC FIBRE SECTION

For the distributed plasticity elements considered, the material nonlinear response can be obtained by integration of the force-deformation response. At any location along the element, the internal section force (stress resultant) can be obtained from the corresponding section deformation (generalised strain) and the constitutive relationship between forces and deformations.

The abstract SectionForceDeformation class computes section forces. One of its subclasses FiberSection3d divides the cross-section into a number of fibre sections so that the distributed plasticity is considered. By doing this, an element can evaluate its equilibrium inside the local system without considering how the constitute response is computed in each section.

Local linear strain is assumed in OpenSees original source code, i.e.

$$\varepsilon^* = u' - yv'' - zw'' \tag{65}$$

Compared with Eq. (37), the missing of second order terms may lead to inaccurate results when torsion effects are significant [9], while the missing warping terms may lead to inaccurate results in analysis of structures with open section.

In this report, a new subclass FiberSectionW3d is created. In the new class, Eq. (37) is used to formulate strain-displacement relationships, leading to the section force vector $\mathbf{D}$ in Eq. (54) to be calculated as,

$$\mathbf{D} = \begin{bmatrix} P \\ M_z \\ M_y \\ W \\ B \\ T_u \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} \sigma_i A_i \\ \sum_{i=1}^{n} -y_i \sigma_i A_i \\ \sum_{i=1}^{n} -z_i \sigma_i A_i \\ \sum_{i=1}^{n} (y_i^2 + z_i^2) \sigma_{iA_i} \\ \sum_{i=1}^{n} -\varpi_i \sigma_i A_i \\ GJ\varphi' \end{bmatrix} \tag{66}$$

where n is the number of fibre sections in arbitrary introduced integration points around the section. The torque due to uniform torsion (T_u) is achieved directly from the first derivative of the twist rotation (φ') and the equivalent tangent modulus (GJ) because of the assumption that shear strain is always elastic, i.e. it is not attempted to consider the presence of combined normal stress and shear stress in determining the inelastic response. For the same reason, only the normal stress (σ) is considered when checking the yielding of the material. The calculation of the sectorial coordinate (ϖ) varies with different section types. The sectorial coordinate of a doubly symmetric I-section is shown in Figure 6.

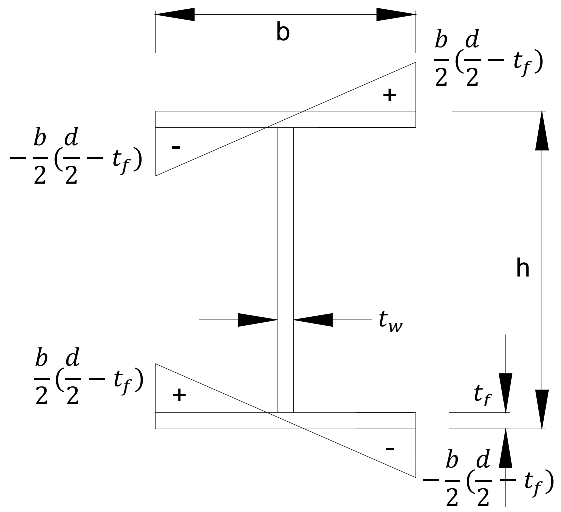

Figure 6 Calculation of sectorial area (ϖ)

The section tangent stiffness matrix $\mathbf{K}$ can be achieved from Eq. (58)

$$\mathbf{K} = \begin{bmatrix} EA & 0 & 0 & EI_p & 0 & 0 \\ 0 & EI_z & 0 & 0 & 0 & 0 \\ 0 & 0 & EI_y & 0 & 0 & 0 \\ EI_p & 0 & 0 & E\alpha & 0 & 0 \\ 0 & 0 & 0 & 0 & EC_w & 0 \\ 0 & 0 & 0 & 0 & 0 & GJ \end{bmatrix} \tag{67}$$

where $EI_p = \int_A E(x^2 + y^2)\, dA$, $E\alpha = \int_A E(x^2 + y^2)^2\, dA$.

INTRODUCING ELASTIC BEAM-COLUMN WARPING ELEMENT INTO OPENSEES

The Elastic Beam-Column Element in OpenSees is implemented in the class of ElasticBeam3d, which is a subclass of the abstract Element class. The functions of the Element class are to compute the tangent stiffness matrix, to compute resisting forces and to store the element state. Displacements and resisting forces are updated during each iteration, until residual quantities (displacements or forces) are less than the specified maximum tolerance value.

Besides the local and global systems, a third basic system is used in OpenSees. The displacement vector $\mathbf{n}_b^*$ in the basic system is defined as follows,

$$\mathbf{n}_b^* = \{e, \theta_{l2}, \theta_{l5}, \theta_{l3}, \theta_{l6}, \theta_{l4} - \theta_{l1}\}^T \tag{68}$$

Compared with Eq. (60), the basic system has only one torsion degree of freedom ($\theta_{l4b} - \theta_{l1b}$). This is because uniform torsion is assumed in OpenSees, and the twist rate along the member has a constant value.

The stiffness matrix $\mathbf{K}_{lb}^*$ and the internal force $\mathbf{p}_b^*$ are calculated based on the basic system and can be transformed to the local system via a transformation matrix $\mathbf{T}_p$. They are expressed as follows,

$$\mathbf{K}_{lb}^{*} = \begin{bmatrix} \frac{EA}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{4EI_z}{L} & \frac{2EI_z}{L} & 0 & 0 & 0 \\ 0 & \frac{2EI_z}{L} & \frac{4EI_z}{L} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{4EI_y}{L} & \frac{2EI_y}{L} & 0 \\ 0 & 0 & 0 & \frac{2EI_y}{L} & \frac{4EI_y}{L} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{GJ}{L} \end{bmatrix} \tag{69}$$

$$\mathbf{p}_b^{*} = \mathbf{K}_{lb}^{*}\mathbf{n}_b^{*} \tag{70}$$

$$\mathbf{T}_p = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{71}$$

so that

$$\mathbf{p}^{*} = \mathbf{T}_p^{\mathrm{T}}\mathbf{p}_b^{*} \tag{72}$$

$$\mathbf{K}_l^{*} = \mathbf{T}_p^{\mathrm{T}}\mathbf{K}_{lb}^{*}\mathbf{T}_p \tag{73}$$

In the present report, to introduce the Elastic Beam-Column Warping Element, a new class ElasticBeamW3d is created. In this class, the basic system is eliminated (because the rate of twist (φ') is no longer constant) and two torsion degrees of freedom are introduced at each node. Eqs. (28) - (36) are used to calculate the stiffness matrix and internal resisting forces in the local system. Since the material is assumed elastic throughout the analysis, no yielding check is necessary. As the cross-section properties are specified in the Tcl files and kept constant during the analysis, class ElasticBeamW3d does not interact with class SectionForceDeformation.

INTRODUCING DISPLACEMENT BASED BEAM-COLUMN WARPING ELEMENT TO OPENSEES

The Displacement Based Beam-Column Element in OpenSees is implemented in the class DispBeamColumn3d, which is a subclass of the abstract Element class.

Strains and stresses are updated and plasticity is accounted for by interacting the Element class with the abstract class SectionForceDeformation during each iteration. In particular, if a fibre section is used, it interacts with subclass Fibersection3D. Once the normal stress of a certain fibre reaches the specified yield stress, the Young's modulus will change in the way specified in the material property. A Gauss-Lobatto integration procedure is used, in which a maximum of 10 integration points can be set along the element, allowing the gradual spread of yielding to be captured. Section weights are introduced in the integrations.

The Displacement Based Beam-Column Element in the original OpenSees code is based on linear strain assumption, in which only axial strain due to elongation, bending and uniform shear strain is accounted for. The local element stiffness matrix $\mathbf{K}_{lb}^{*}$ is expressed as,

$$\mathbf{K}_{lb}^{*} = \begin{bmatrix} \sum_{i=1}^{n_2} \frac{EA}{L} W_t(i) & 0 & 0 & 0 & 0 & 0 \\ 0 & \sum_{i=1}^{n_2} \frac{\left(\frac{6x_i}{L}-4\right)^2}{L^2} EI_z W_t(i) & \sum_{i=1}^{n_2} \frac{\left(\frac{6x_i}{L}-4\right)\left(\frac{6x_i}{L}-2\right)}{L^2} EI_z W_t(i) & 0 & 0 & 0 \\ 0 & \sum_{i=1}^{n_2} \frac{\left(\frac{6x_i}{L}-4\right)\left(\frac{6x_i}{L}-2\right)}{L^2} EI_z W_t(i) & \sum_{i=1}^{n_2} \frac{\left(\frac{6x_i}{L}-2\right)^2}{L^2} EI_z W_t(i) & 0 & 0 & 0 \\ 0 & 0 & 0 & \sum_{i=1}^{n_2} \frac{\left(\frac{6x_i}{L}-4\right)^2}{L^2} EI_y W_t(i) & \sum_{i=1}^{n_2} \frac{\left(\frac{6x_i}{L}-4\right)\left(\frac{6x_i}{L}-2\right)}{L^2} EI_y W_t(i) & 0 \\ 0 & 0 & 0 & \sum_{i=1}^{n_2} \frac{\left(\frac{6x_i}{L}-4\right)\left(\frac{6x_i}{L}-2\right)}{L^2} EI_y W_t(i) & \sum_{i=1}^{n_2} \frac{\left(\frac{6x_i}{L}-2\right)^2}{L^2} EI_y W_t(i) & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{GJ}{L} \end{bmatrix} \quad (74)$$

where x_i is the longitudinal coordinate of the integration point, n_2 is the number of integration points along the member and $W_t(i)$ is the weighting of each integration point.

In the present report, a new class DispBeamColumnW3d is created to introduce the Displacement Based Beam-Column Warping Element. In this class, a second order approximation of the Green-Lagrange strains is used to formulate the local element stiffness matrix and resisting forces. New matrices $\mathbf{N}_{\delta d1}$, $\mathbf{N}_{\delta d2}$ and $\mathbf{G}$ are introduced to handle the corresponding terms related to second order strains. Eq. (37) is used to check if the strain exceeds the yielding strain. As OpenSees provides a number of matrix and vector operation functions, the stiffness matrix $\mathbf{K}_l$ and resisting force $\mathbf{p}_l$ are calculated through matrix multiplication as per Eq. (54) and Eq. (56).

BENCHMARK PROBLEMS

ELASTIC LATERAL BUCKLING OF A CANTILEVER RIGHT-ANGLED FRAME UNDER END-LOAD

This example involves the response of a right-angled frame as shown in Figure 7. A point load P is applied in z-direction at the element tip and a perturbation load ($P_r = 0.0002P$) is applied in the x-direction. Young's modulus is taken as 491,182 MPa (71,240 ksi). The cross-section is rectangular and the dimension is shown in Figure 7. This problem has previously been analysed by Crisfield [11] and Alemdar [17]. The frame is modelled with four elements, two in each leg. The problem is solved using the Elastic Beam-Column Warping Element, and a displacement controlled iterative solution procedure is employed. The frame is unaffected by warping effects since the cross-section is solid. Accordingly, warping restraints are released at the fixed end and at the beam-column connection of the frame. The computed response between the applied load and the tip lateral deflection is shown in Figure 8, where the results are shown to compare well with Elastic Beam-Column element results in OpenSees (without the warping degree of freedom) and results from Alemdar [17]. The example shows that the present formulation is also applicable to elements without warping effects. It should be noted that the three analyses use the same number of elements.

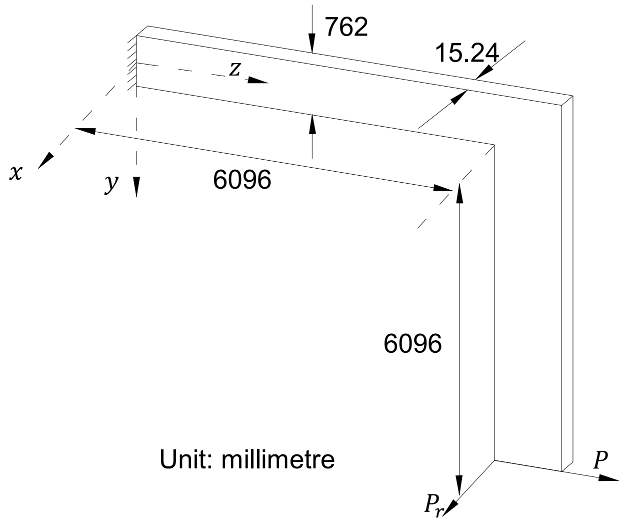

Figure 7 Initial geometry for right-angled frame

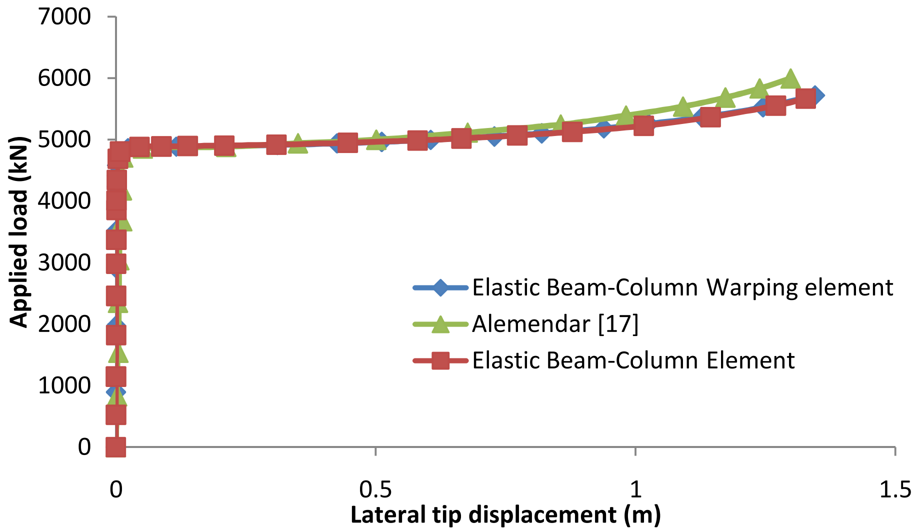

Figure 8 Load-tip displacement for right-angled frame

CANTILEVER BEAM SUBJECTED TO END TORQUE

The example considered here is a cantilever beam subjected to end torque as shown in Figure 9. Three different boundary conditions are considered: (a) warping free at both ends of the beam; (b) warping restrained at the fixed end of the beam and warping free at the free end of the beam; (c) warping restrained at both ends of the beam. The geometry and material properties are $L = 5.096$ m (240 in.), $b = 213.9$mm (8.42 in.), $t_f = 23.6$ mm (0.93 in.), $d = 549.1$mm (21.62 in.), $t_w = 14.7$ mm (0.58 in.), Young's modulus $E = 2 \times 10^5$ MPa (29,000 ksi), and shear modulus $G = 77{,}221$ MPa ($11{,}200$ ksi). This problem has previously been analysed by Lin and Hsiao [8]. The problem is solved using both the Elastic Beam-Column Warping Element and the Displacement Based Beam-Column Warping Element. Examples of Tcl script files for the two elements for case (a) are listed in Appendix 1 and Appendix 2. Twenty elements are used along the member. The relationships between the end torque and end twist rotation for the three cases are shown in Figure 10, Figure 11 and Figure 12, and compared with results from the literature. As can be seen the three curves agree well when the twist angle is small. The Elastic Beam-Column Warping Element results exhibit linear relations between the angle and the torque, while the other two elements show that the torsional stiffness increases with increasing angle of twist. This indicates the necessity of using second order strains when the twist angle is not small. It is concluded that the Elastic Beam-Column Warping Element may give inaccurate results when the torsional effects are important.

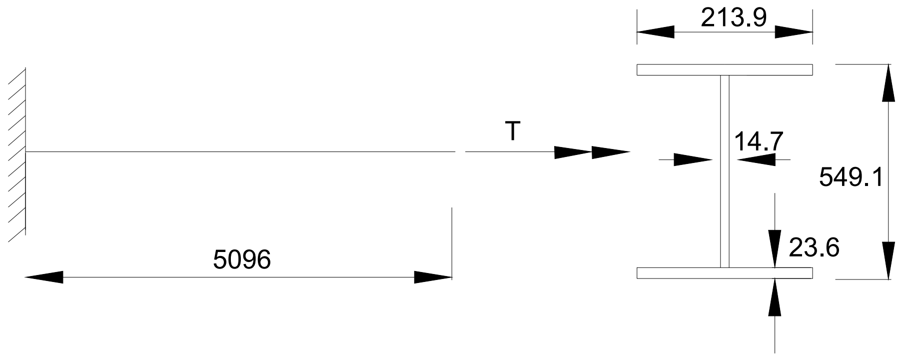

Figure 9 Cantilever subject to end torque

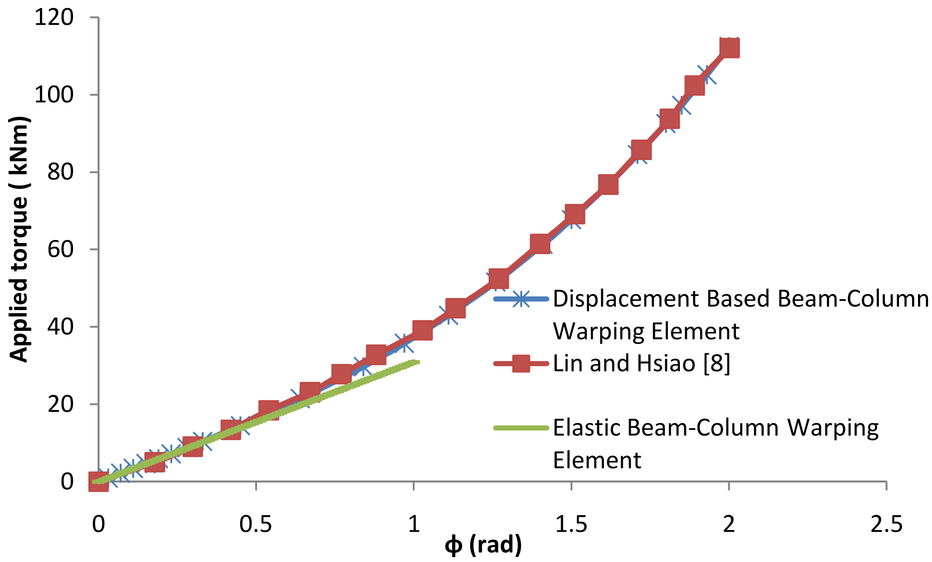

Figure 10 Load-end twist angle for Cantilever beam (case (a))

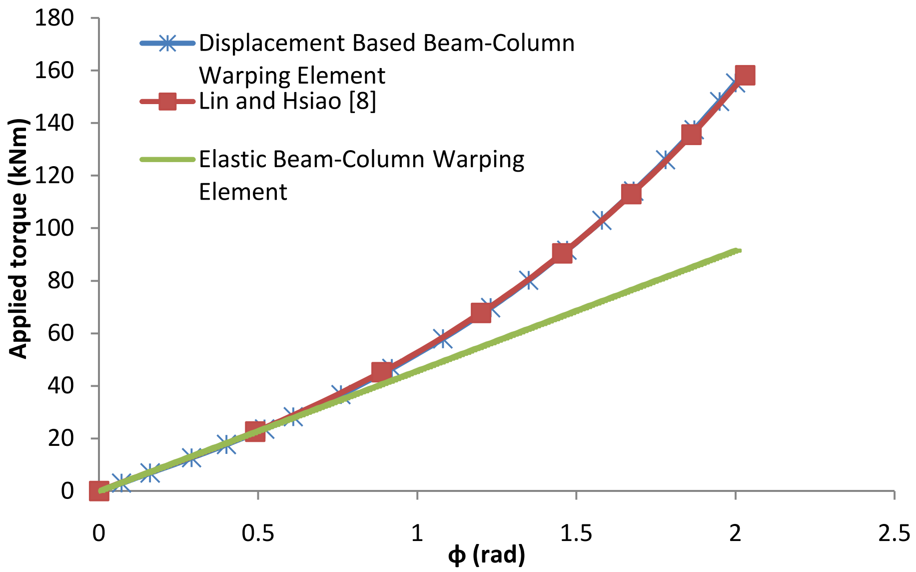

Figure 11 Load-end twist angle for Cantilever beam (case (b))

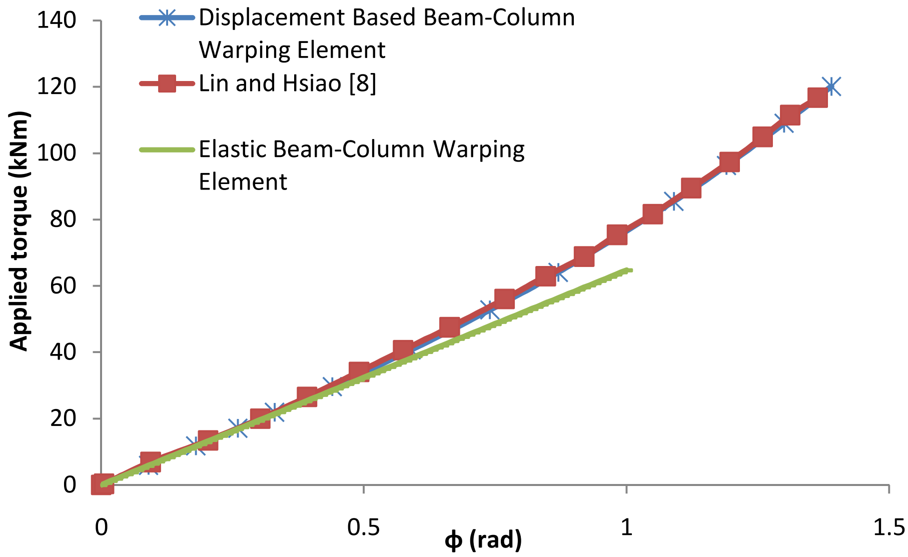

Figure 12 Load-end twist angle for Cantilever beam (case (c))

ELASTIC LATERAL BUCKLING OF AN I-BEAM IN PURE BENDING

A W10x100 beam is subjected to pure bending about its major axis, and a small perturbation moment at one end about its minor axis, as shown in Figure 13. The geometry and material properties are $L = 5.096$ m (240 in.), $b = 262.6$ mm (10.34 in.), $t_f = 28.4$ mm (1.12 in.), $d = 282.2$ mm (11.11 in.), $t_w = 17.3$ mm (0.68 in.), Young's modulus $E = 2 \times 10^5$ MPa (29,000 ksi), and shear modulus $G = 77{,}221$ MPa ($11{,}200$ ksi). The beam is simply supported at both ends and twisting and warping are restrained at its ends. Both Elastic Beam-Column Warping Element and Displacement Based Beam-Column Warping Element are used to model this problem. Four elements are used along the beam. The same problem has been modelled by Alemdar [17]. Figure 14 shows the applied moment versus central out-of-plane relationships for the two elements and a comparison with results obtained using the software FE++ [22], where element B14Disp (displacement based beam which accounts for warping effects) is used. It is found that the three elements predict similar beam behaviour while the deflection is moderate but a discrepancy can be seen between the Elastic Beam-Column Warping Element and the other two elements at large lateral deflections.

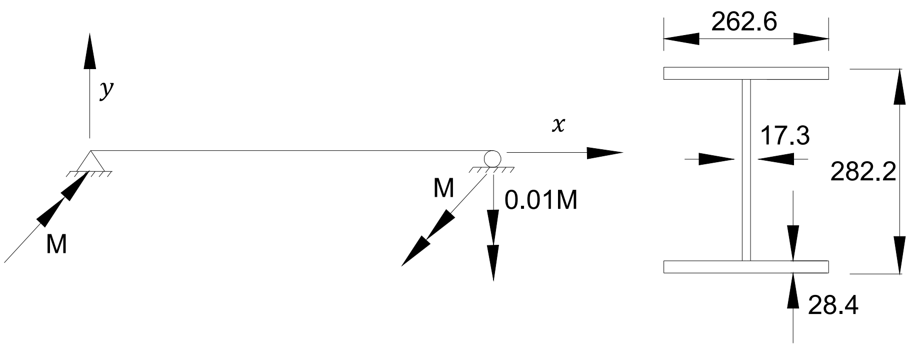

Figure 13 Lateral buckling of an I-beam

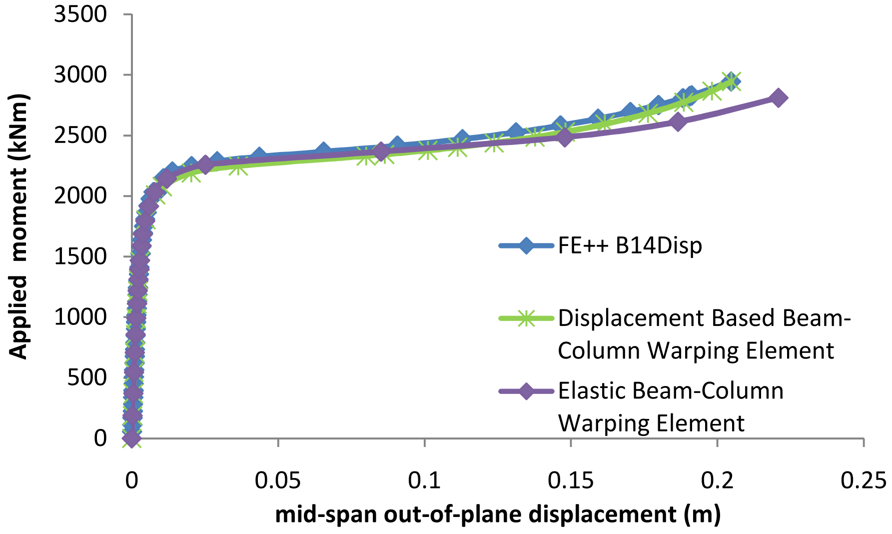

Figure 14 Moment-mid-plane displacement for I-beam

INELASTIC LATERAL BUCKLING OF A CANTILEVER BEAM UNDER VERTICAL LOAD

The cantilever shown in Figure 15 is subjected to a vertical load at its tip, and is fully restrained against warping at the support. The geometry and material properties are $L = 10$ m, $b = 190$ mm, $t_f = 13$ mm, $d = 613$mm, $t_w = 25$ mm, Young's modulus $E = 20{,}600$ MPa, shear modulus $G = 7{,}923$ MPa, and yield stress $\sigma_y = 206$ MPa. A perfect elastic-plastic material response is assumed. This problem has previously been modelled by Izzuddin and Smith [23].

The theoretical elastic lateral torsional buckling load (P_{cr}) is given by

$$P_{cr} = \frac{\sqrt{EI_y GJ}}{L^2}\left(3.95 + 3.52\sqrt{\frac{\pi^2 EC_w}{GJL^2}}\right) = 47.35 \text{ kN} \tag{75}$$

The cantilever is modelled using the Displacement Based Beam-Column Warping Element. Ten elements are modelled along the length, and a small perturbation lateral force is introduced to initiate out-of-plane buckling. The non-dimensional vertical load versus the lateral tip displacement is shown in Figure 16. The result demonstrates the element's ability to predict elastic buckling load with high accuracy. The post-ultimate curve also agrees well with the literature.

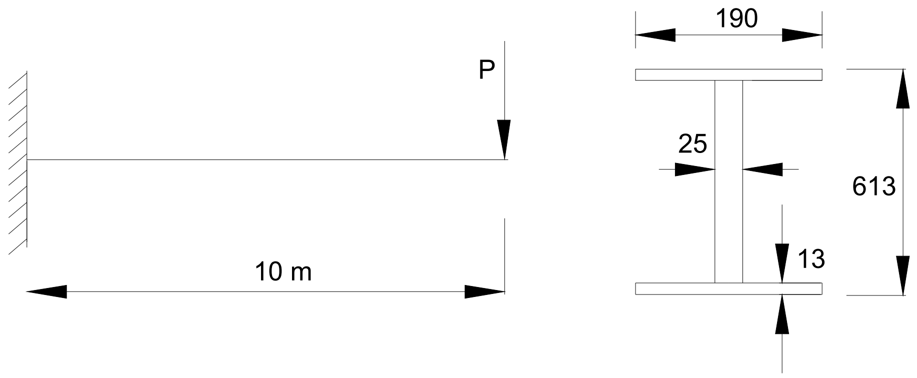

Figure 15 Cantilever beam subject to end point load

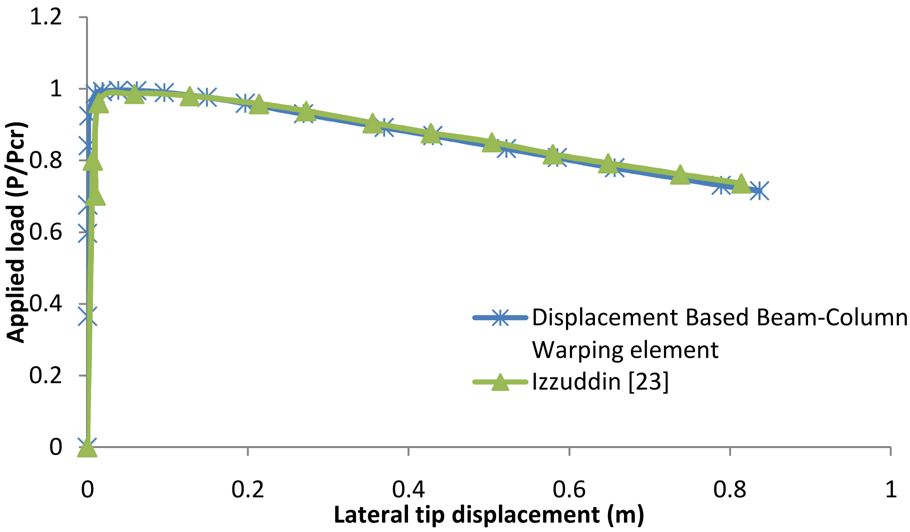

Figure 16 Load-tip displacement for Cantilever beam

INELASTIC LATERAL BUCKLING OF AN I-BEAM IN PURE BENDING

A W10x100 beam is subjected to pure bending about its major axis and a small perturbation moment at one end about its minor axis, i.e. the problem definition is the same as that in Section 'ELASTIC LATERAL BUCKLING OF AN I-BEAM IN PURE BENDING' except that the material is inelastic. The yield stress is $\sigma_y = 344.7$ MPa $(50$ ksi) and the stress-strain relationship is assumed to be elastic linearly hardening with a stiffness of 1500 MPa (217.5 ksi) in the inelastic hardening range. The element is modelled with four Displacement Based Beam-Column Warping Elements. The same problem has been analysed by Alemdar [17]. The results of the relationship between the applied load and mid-span out-of-plane displacement are compared with Abaqus [24] results where twelve B31OS elements are used, as shown in Figure 17. It should be noticed that both elements include the warping degree of freedom. It is observed that the two programs produce almost identical results.

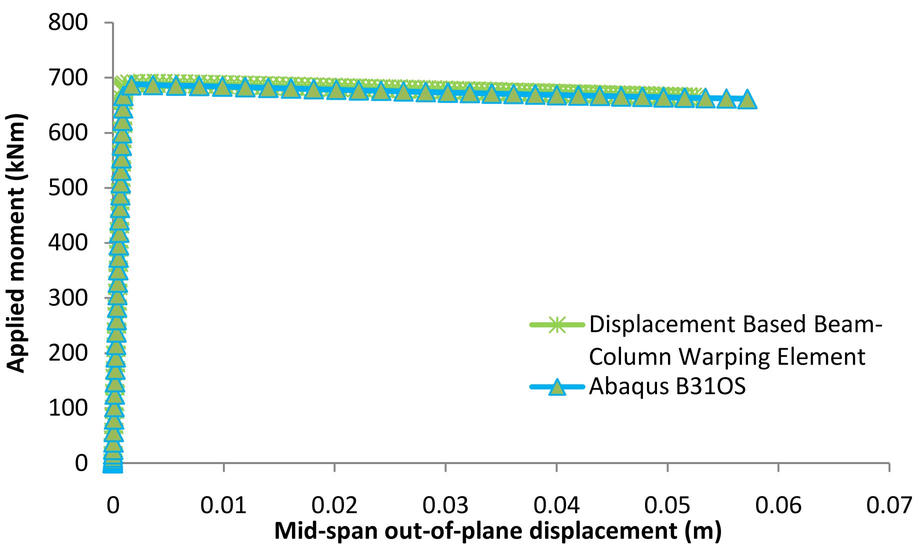

Figure 17 Moment-mid-plane displacement for I-beam

CONCLUSION

The implementation and application of three-dimensional analyses for doubly symmetric beam-columns with warping effects in software package OpenSees are presented. A co-rotational framework is utilized based on the works of Crisfield [11, 16] and Alemdar [17]. Two different elements, Elastic Beam-Column Warping Element and Displacement Based Beam-Column Warping Element, are incorporated into OpenSees. Abstract classes ModelBuilder, Element, GeometricTransformation and SectionForceDeformation in OpenSees are modified. In particular, the restrictions related to number of degrees of freedom for three-dimensional analysis are released.

Several examples of single beams and simple frames subject to flexural-torsional buckling have been presented. The results have been compared with published papers and program packages ABAQUS and FE++ to demonstrate the accuracy of the present work. The comparisons show that the Displacement Based Beam-Column Warping Element is accurate at large displacements and rotations, while the Elastic Beam-Column Warping Element is accurate at moderate displacements and rotations but becomes inaccurate at large deformations when torsion effects are significant.

REFERENCES

1. Pi, Y.L. and Trahair, N.S., *Nonlinear Inelastic Analysis of Steel Beam-Columns. I: Theory.* Journal of Structural Engineering, 1994. **120**(7): p. 2041-2061.

2. Pi, Y.L. and Trahair, N.S., *Nonlinear Inelastic Analysis of Steel Beam-Columns. II: Applications.* Journal of Structural Engineering, 1994. **120**(7): p. 2062-2085.

3. Kwak, H.-G., Kim, D.-Y., and Lee, H.-W., *Effect of warping in geometric nonlinear analysis of spatial beams.* Journal of Constructional Steel Research, 2001. **57**(7): p. 729-751.

4. Pai, P.F., Anderson, T.J., and Wheater, E.A., *Large-deformation tests and total-Lagrangian finite-element analyses of flexible beams.* International Journal of Solids and Structures, 2000. **37**(21): p. 2951-2980.

5. Conci, A. and Gattass, M., *Natural approach for thin-walled beam-columns with elastic-plasticity.* International Journal for Numerical Methods in Engineering, 1990. **29**(8): p. 1653-1679.

6. Hsiao, K.-M., Hong, H.-J., and Chen, Y.-R., *A corotational procedure that handles large rotations of spatial beam structures.* Computers & Structures, 1987. **27**(6): p. 769-781.

7. Hsiao, K.M., Lin, J.Y., and Lin, W.Y., *A consistent co-rotational finite element formulation for geometrically nonlinear dynamic analysis of 3-D beams.* Computer Methods in Applied Mechanics and Engineering, 1999. **169**(1-2): p. 1-18.

8. Lin, W.Y. and Hsiao, K.M., *Co-rotational formulation for geometric nonlinear analysis of doubly symmetric thin-walled beams.* Computer Methods in Applied Mechanics and Engineering, 2001. **190**(45): p. 6023-6052.

9. Battini, J.-M. and Pacoste, C., *Co-rotational beam elements with warping effects in instability problems.* Computer Methods in Applied Mechanics and Engineering, 2002. **191**(17-18): p. 1755-1789.

10. Alemdar, B.N. and White, D.W., *Displacement, Flexibility, and Mixed Beam--Column Finite Element Formulations for Distributed Plasticity Analysis.* Journal of Structural Engineering, 2005. **131**(12): p. 1811-1819.

11. Crisfield, M.A., *A consistent co-rotational formulation for non-linear, three-dimensional, beam-elements.* Computer Methods in Applied Mechanics and Engineering, 1990. **81**(2): p. 131-150.

12. Teh, L.H. and Clarke, M.J., *Co-rotational and Lagrangian formulations for elastic three-dimensional beam finite elements.* Journal of Constructional Steel Research, 1998. **48**(2-3): p. 123-144.

13. Oran, C., *Tangent stiffness in plane frames.* Journal of the Structural Division., ASCE, 1973. **99**(ST6): p. 973-985.

14. Oran, C., *Tangent stiffness in space frames.* Journal of the Structural Division., ASCE, 1973. **99**(ST6): p. 987-1001.

15. McKenna, F., Fenves, G.L., and Scott, M.H., *"Open system for earthquake engineering simulation."*. 2000, Univ. of California, Berkeley, Calif., <http://opensees. berkeley. edu>.

16. Crisfield, M.A., *Non-linear finite element analysis of solids and structures, Volume2: Advanced topics*. 1997, Baffins Lane, Chichester, West Sussex PO19 1UD, England: John Wiley & Sons Ltd.

17. Alemdar, B.N., *Distributed Plasticity Analysis of Steel Building Structural Systems.* PhD thesis, Georgia Institute of Technology, 2001.

18. Argyris, J., *An excursion into large rotations.* Computer Methods in Applied Mechanics and Engineering, 1982. **32**(1-3): p. 85-155.

19. Rasmussen, K.J.R., *Bifurcation of locally buckled members.* Thin-Walled Structures, 1997. **28**(2): p. 117-154.

20. Fenves, G.L., et al. *An object-oriented software environment for collaborative network simulation*. in *Paper No. 1492, Proceedings, 13World Conference on Earthquake Engineering*. 2004.

21. Scott, M.H., et al., *Software Patterns for Nonlinear Beam-Column Models.* Journal of Structural Engineering, 2008. **134**(4): p. 562-571.

22. Lu, J., et al., *an object-oriented application framework for finite element programming*, in *The Second Annual Object-Oriented Numerics Conference*. 1994: Sunriver, Oregon, USA. p. 438-447.

23. Izzuddin, B.A. and Smith, D.L., *Large-Displacement Analysis of Elastoplastic Thin-Walled Frames. II: Verification and Application.* Journal of Structural Engineering, 1996. **122**(8): p. 915-925.

24. ABAAQUS, *Inc. ABAQUS. Ver. 6.8. analysis user's manual*. 2009.

APPENDIX 1: TCL SCRIPT FOR CANTILEVER BEAM WITH END TORQUE (ELASTIC BEAM-COLUMN WARPING ELEMENT)

```
# -----------------------------------------------------------------------------------
-------------------
# Appendix 1. Cantilever beam subject to end torque (Elastic Beam-Column Warping
Element)

# SET UP ----------------------------------------------------------------------------
----
wipe;                                # clear opensees model
source Wsection.tcl                  # include definition for I-section
source LibUnits.tcl                  # include units
model basic -ndm 3 -ndf 7;           # 3 dimensions, 7 dof per node
set dir Cantilever-endtorque
file mkdir $dir;                     # create data directory

# define GEOMETRY -------------------------------------------------------------
# nodal coordinates:
node 1 0 0 0;
node 2 12 0 0                        # node#, X Y z
node 3 24 0 0
node 4 36 0 0
node 5 48 0 0
node 6 60 0 0
node 7 72 0 0
node 8 84 0 0
node 9 96 0 0
node 10 108 0 0
node 11 120 0 0
node 12 132 0 0
node 13 144 0 0
node 14 156 0 0
node 15 168 0 0
node 16 180 0 0
node 17 192 0 0
node 18 204 0 0
node 19 216 0 0
node 20 228 0 0
node 21 240 0 0
# Single point constraints -- Boundary Conditions
fix 1 1 1 1 1 1 1 0;

# define material and section
set poisson 0.3
set G 11200.0;
set J 5.861
set GJ [expr $G*$J]
set Cw 9902.0;
set E 29000.0
set A 27.3
set Iz 2070.0
set Iy 92.9
#W21x93 section

# define geometric transformation: performs a corotational geometric
transformation of beam stiffness and resisting force from the basic system to the
global-coordinate system
set ColTransfTag 1;
geomTransf Corotational $ColTransfTag 0 0 1;          # associate a tag to column
transformation
```

```
# Define ELEMENTS --------------------------------------------------------------
element elasticBeamColumn 1 1 2 $A $E $G $J $Iy $Iz $ColTransfTag $Cw;
element elasticBeamColumn 2 2 3 $A $E $G $J $Iy $Iz $ColTransfTag $Cw;
element elasticBeamColumn 3 3 4 $A $E $G $J $Iy $Iz $ColTransfTag $Cw;
element elasticBeamColumn 4 4 5 $A $E $G $J $Iy $Iz $ColTransfTag $Cw;
element elasticBeamColumn 5 5 6 $A $E $G $J $Iy $Iz $ColTransfTag $Cw;
element elasticBeamColumn 6 6 7 $A $E $G $J $Iy $Iz $ColTransfTag $Cw;
element elasticBeamColumn 7 7 8 $A $E $G $J $Iy $Iz $ColTransfTag $Cw;
element elasticBeamColumn 8 8 9 $A $E $G $J $Iy $Iz $ColTransfTag $Cw;
element elasticBeamColumn 9 9 10 $A $E $G $J $Iy $Iz $ColTransfTag $Cw;
element elasticBeamColumn 10 10 11 $A $E $G $J $Iy $Iz $ColTransfTag $Cw;
element elasticBeamColumn 11 11 12 $A $E $G $J $Iy $Iz $ColTransfTag $Cw;
element elasticBeamColumn 12 12 13 $A $E $G $J $Iy $Iz $ColTransfTag $Cw;
element elasticBeamColumn 13 13 14 $A $E $G $J $Iy $Iz $ColTransfTag $Cw;
element elasticBeamColumn 14 14 15 $A $E $G $J $Iy $Iz $ColTransfTag $Cw;
element elasticBeamColumn 15 15 16 $A $E $G $J $Iy $Iz $ColTransfTag $Cw;
element elasticBeamColumn 16 16 17 $A $E $G $J $Iy $Iz $ColTransfTag $Cw;
element elasticBeamColumn 17 17 18 $A $E $G $J $Iy $Iz $ColTransfTag $Cw;
element elasticBeamColumn 18 18 19 $A $E $G $J $Iy $Iz $ColTransfTag $Cw;
element elasticBeamColumn 19 19 20 $A $E $G $J $Iy $Iz $ColTransfTag $Cw;
element elasticBeamColumn 20 20 21 $A $E $G $J $Iy $Iz $ColTransfTag $Cw;

# Define RECORDERS -------------------------------------------------------------
recorder Node -file $dir/DFree.out -time -node 21 -dof 1 2 3 4 5 6 7 disp;
            # record displacements of the free node

# define end torque -------------------------------------------------------------
pattern Plain 1 Linear {
  load 21 0.0 0.0 0.0 1000.0 0.0 0.0 0.0
}

constraints Plain;                          # how it handles boundary conditions

numberer Plain;                             # renumber dof's to minimize band-width
(optimization), if you want to

system BandGeneral;                         # how to store and solve the system of
equations in the analysis

test NormDispIncr 1.0e-8 10 ;                           # determine if convergence
has been achieved at the end of an iteration step

algorithm NewtonLineSearch;                             # use Newton line search

integrator DisplacementControl 21 4 0.01;                     # determine the time
step for an analysis

analysis Static                             # define type of analysis static or
transient

analyze 100;                                # perform analysis

puts "Done!"
```

APPENDIX 2: TCL SCRIPT FOR CANTILEVER BEAM WITH END TORQUE (DISPLACEMENT BASED BEAM-COLUMN WARPING ELEMENT)

```
# ------------------------------------------------------------------------------------------------------------------
# Appendix 2. Cantilever beam subject to end torque (Displacement Based Beam-
Column Warping Element)

# SET UP ------------------------------------------------------------------------------------------
wipe;                                    # clear opensees model
source Wsection.tcl;                      # include definition for I-section
source LibUnits.tcl;                      # include units
model basic -ndm 3 -ndf 7;               # 3 dimensions, 7 dof per node
set dir Cantilever-endtorque
file mkdir $dir;                         # create data directory

# define GEOMETRY -----------------------------------------------------------
# nodal coordinates:
node 1 0 0 0;
node 2 12 0 0                            # node#, X Y z
node 3 24 0 0
node 4 36 0 0
node 5 48 0 0
node 6 60 0 0
node 7 72 0 0
node 8 84 0 0
node 9 96 0 0
node 10 108 0 0
node 11 120 0 0
node 12 132 0 0
node 13 144 0 0
node 14 156 0 0
node 15 168 0 0
node 16 180 0 0
node 17 192 0 0
node 18 204 0 0
node 19 216 0 0
node 20 228 0 0
node 21 240 0 0

# Single point constraints -- Boundary Conditions
fix 1 1 1 1 1 1 1 0;

# define material and section
set IDsteel 1;                           # assign material tag
set Fy 500000;                           # assign a super large yielding stress
set Es 29000.0;
set Bs 0.000001;                         # strain-hardening ratio
set R0 15;
set poisson 0.3
set G 11200.0;
set J 5.861
set GJ [expr $G*$J]
set BeamSecTagFiber 1;                  # assign a tag number to the beam section
fiber

set SecTagTorsion 70;                    # assign a tag number to the torsion
information of the beam
set BeamSecTag 3
```

```
uniaxialMaterial Steel02 $IDsteel $Fy $Es $Bs;                   # build steel01
material

set d [expr 21.62*$in]; # depth
      set bf [expr 8.42*$in]; # flange width
      set tf [expr 0.93*$in]; # flange thickness
      set tw [expr 0.58*$in]; # web thickness
      set nfdw 32;            # number of fibers along dw
      set nftw 4;       # number of fibers along tw
      set nfbf 32;            # number of fibers along bf
      set nftf 4;             # number of fibers along tf

Wsection $BeamSecTagFiber $IDsteel $d $bf $tf $tw $nfdw $nftw $nfbf $nftf;
#build beam section

uniaxialMaterial Elastic $SecTagTorsion $GJ

section Aggregator $BeamSecTag $SecTagTorsion T -section $BeamSecTagFiber;   #
add elastic torsion

set numIntgrPts 5;    # number of integration points for dis-based element
set BeamTransfTag 1;    # associate a tag to beam transformation
geomTransf Corotational $BeamTransfTag 0 0 1

# define geometric transformation: performs a corotational geometric
transformation of beam stiffness and resisting force from the basic system to the
global-coordinate system

# Define ELEMENTS ----------------------------------------------------------------
element dispBeamColumn 1 1 2 $numIntgrPts $BeamSecTag $BeamTransfTag
element dispBeamColumn 2 2 3 $numIntgrPts $BeamSecTag $BeamTransfTag
element dispBeamColumn 3 3 4 $numIntgrPts $BeamSecTag $BeamTransfTag
element dispBeamColumn 4 4 5 $numIntgrPts $BeamSecTag $BeamTransfTag
element dispBeamColumn 5 5 6 $numIntgrPts $BeamSecTag $BeamTransfTag
element dispBeamColumn 6 6 7 $numIntgrPts $BeamSecTag $BeamTransfTag
element dispBeamColumn 7 7 8 $numIntgrPts $BeamSecTag $BeamTransfTag
element dispBeamColumn 8 8 9 $numIntgrPts $BeamSecTag $BeamTransfTag
element dispBeamColumn 9 9 10 $numIntgrPts $BeamSecTag $BeamTransfTag
element dispBeamColumn 10 10 11 $numIntgrPts $BeamSecTag $BeamTransfTag
element dispBeamColumn 11 11 12 $numIntgrPts $BeamSecTag $BeamTransfTag
element dispBeamColumn 12 12 13 $numIntgrPts $BeamSecTag $BeamTransfTag
element dispBeamColumn 13 13 14 $numIntgrPts $BeamSecTag $BeamTransfTag
element dispBeamColumn 14 14 15 $numIntgrPts $BeamSecTag $BeamTransfTag
element dispBeamColumn 15 15 16 $numIntgrPts $BeamSecTag $BeamTransfTag
element dispBeamColumn 16 16 17 $numIntgrPts $BeamSecTag $BeamTransfTag
element dispBeamColumn 17 17 18 $numIntgrPts $BeamSecTag $BeamTransfTag
element dispBeamColumn 18 18 19 $numIntgrPts $BeamSecTag $BeamTransfTag
element dispBeamColumn 19 19 20 $numIntgrPts $BeamSecTag $BeamTransfTag
element dispBeamColumn 20 20 21 $numIntgrPts $BeamSecTag $BeamTransfTag

# Define RECORDERS ---------------------------------------------------------------
recorder Node -file $dir/DFree.out -time -node 21 -dof 1 2 3 4 5 6 7 disp;
            # record displacements the free node

# define end torque ---------------------------------------------------------------
pattern Plain 1 Linear {
  load 21 0.0 0.0 0.0 1000.0 0.0 0.0 0.0
}

constraints Plain;                          # how it handles boundary conditions
```

```
numberer Plain;                          # renumber dof's to minimize band-width
(optimization), if you want to

system BandGeneral;                      # how to store and solve the system of
equations in the analysis

test NormDispIncr 1.0e-8 10 ;                        # determine if convergence
has been achieved at the end of an iteration step

algorithm NewtonLineSearch;                          # use Newton line search

algorithm: updates tangent stiffness at every iteration

integrator DisplacementControl 21 4 0.01;                  # determine the time
step
analysis Static;                         # define type of analysis static or
transient
analyze 100;                             # perform analysis

puts "Done!"
```