Primljen / Received: 9.10.2015.
Ispravljen / Corrected: 16.2.2016.
Prihvaćen / Accepted: 19.3.2016.
Dostupno online / Available online: 10.6.2016.

Computer-based analysis of spatial frames according to second order theory

Authors:

Assist.Prof. Ljiljana Žugić, PhD. CE
University of Montenegro
Faculty of Civil Engineering
ljiljaz@ac.me

Prof. Stanko Brčić, PhD. CE
University of Belgrade
Faculty of Civil Engineering
stanko@grf.bg.ac.rs

Prof. Špiro Gopčević, PhD. CE
High railway School of Professional Studies
sgopcevic@yahoo.com

Preliminary report

Ljiljana Žugić, Stanko Brčić, Špiro Gopčević

Computer-based analysis of spatial frames according to second order theory

The analysis of spatial frames formed of beam elements according to the second order theory is presented in the paper. The stiffness matrix and equivalent load vector elements are derived from the exact solution of the corresponding differential equations according to the second order theory. An appropriate computer code ALIN was developed in order to enable numerical formulation of the problem. The code uses the exact stiffness matrix and the exact vector of equivalent loads, as opposed to commercial codes that use the geometric stiffness matrix.

Key words:
linear spatial frames, second order theory, spatial beam element, stiffness matrix, vector of equivalent loads

Prethodno priopćenje

Ljiljana Žugić, Stanko Brčić, Špiro Gopčević

Programska realizacija proračuna prostornih linijskih nosača prema teoriji drugog reda

U radu je prikazana analiza prostornih linijskih nosača, sastavljenih od grednih elemenata, po teoriji drugog reda. Komponente matrice krutosti i vektora ekvivalentnog opterećenja dobivene su na temelju točnoga rješenja odgovarajućih diferencijalnih jednadžbi teorije drugog reda. Radi numeričke realizacije ovoga problema, razvijen je i odgovarajući računalni program ALIN koji se koristi točnom matricom krutosti i točnim vektorom opterećenja, za razliku od komercijalnih programa koji se koriste geometrijskom matricam krutosti.

Ključne riječi:
linijski nosači u prostoru, teorija drugog reda, gredni element u prostoru, matrica krutosti, vektor ekvivalentnog opterećenja

Vorherige Mitteilung

Ljiljana Žugić, Stanko Brčić, Špiro Gopčević

Programmierung des Berechnungsverfahrens für räumliche Stabtragwerke nach der Theorie 2. Ordnung

In dieser Arbeit wird die Analyse räumlicher Stabtragwerke, die aus Balkenelementen zusammengesetzt sind, nach der Theorie 2. Ordnung dargestellt. Die Komponenten der Steifigkeitsmatrix und des Vektors äquivalenter Lasten wurden aufgrund der genauen Lösung der entsprechenden Differentialgleichungen 2. Ordnung hergeleitet. Um das Problem numerisch zu lösen, wurde das Computerprogramm ALIN entwickelt, in dem die genaue Steifigkeitsmatrix und der genaue Lastvektor implementiert sind, im Gegensatz zu kommerziellen Programmen, die geometrische Steifigkeitsmatrizen verwenden.

Schlüsselwörter:
räumliche Stabtragwerke, Theorie 2. Ordnung, räumliche Balkenelemente, Steifigkeitsmatrix, Vektor äquivalenter Lasten

1. Introduction

Basic assumptions of the first order theory are: small deformations, small displacements of action points of external forces, and linear stress-strain relationship. In the second order theory the second assumption is discarded, but the first and the third assumptions are retained, and so equilibrium conditions are established at deformed elements of the structure. It is also assumed that the load is conservative, i.e. that it does not change its direction and intensity during deformation, and also that it is defined as distributed over the undeformed length of the structure.

The paper briefly presents derivation of the stiffness matrix and equivalent load vector according to the second order theory of the spatial beam finite element with two nodal points and six degrees of freedom at each node. The stiffness matrix and equivalent load vector elements due to transverse load along the element are obtained from the exact solution of the corresponding differential equations related to the second order theory of a transversally loaded beam. The implementation of the exact stiffness matrix and exact equivalent load vector enabled development of the computer code ALIN, which was created as a means to analyse spatial, and of course planar, framed structures according to the second order theory.

Up to now, not a single standard FEM-based commercial computer program for analysis of engineering structures has used the exact stiffness matrix and exact equivalent load vector. Rather, their analysis of the second order theory problems is based upon the geometric stiffness matrix obtained from linear theory (differential equation according to the first order theory), i.e. obtained from approximate solution of differential equations of the second order theory. Consequently, in order to obtain more accurate results, it is necessary to assume several finite elements along a single beam, as opposed to the code ALIN in which each beam is modelled as a single finite element.

2. Analysis of spatial beam element

Figure 1 presents a straight beam finite element in 3D space, characterized by the length L, two nodal points, and an arbitrary cross section that is constant along the length of the element.

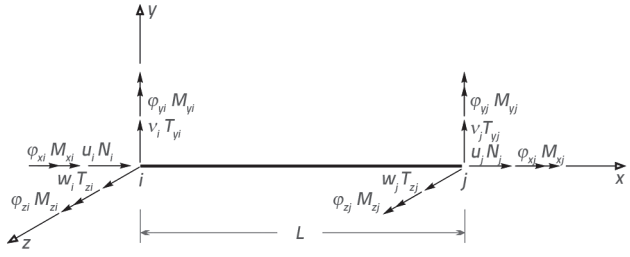


Figure 1. Generalized forces and generalized displacements at nodes of the beam finite element

Generalized coordinates at the spatial beam finite element nodes are displacements *u*, *v*, *w* in the direction of axes x, y, z, and rotations φ_x, φ_y, φ_z about the same axes. Generalized forces (normal force N, transverse forces T_y and T_z, bending moments M_y and M_z and torsional moment M_x) correspond to generalized displacements at element ends. The convention of the positive sense of generalized displacements and generalized forces is presented in Figure 1. Generalized displacements and generalized forces at nodes *i* and *j* constitute the components of the vector of generalized displacements **q** and the vector of generalized forces **R**, respectively:

$$q^T = [u_i\ v_i\ w_i\ \varphi_{xi}\ \varphi_{yi}\ \varphi_{zi}\ u_j\ v_j\ w_j\ \varphi_{xj}\ \varphi_{yj}\ \varphi_{zj}] \quad (1)$$

$$R^T = [N_i\ T_{yi}\ T_{zi}\ M_{xi}\ M_{yi}\ M_{zi}\ N_j\ T_{yj}\ T_{zj}\ M_{xj}\ M_{yj}\ M_{zj}] \quad (2)$$

2.1. Stiffness matrix

As it is known, the relationship between the generalized force vector **R** and the generalized displacement vector **q** is established by the stiffness matrix **k** of the element:

$$\mathbf{R} = \mathbf{k} \cdot \mathbf{q} \quad (3)$$

The stiffness matrix is the order-twelve symmetric square matrix that may be derived using the principle of superposition, or by separating the spatial state of stress of the element into: axial stress, bending in xy plane (about z axis), bending in xz plane (about y axis), and torsion. Therefore, expression (3) may be presented as:

$$\begin{bmatrix} \mathbf{R}_a \\ \mathbf{R}_{sz} \\ \mathbf{R}_{sy} \\ \mathbf{R}_t \end{bmatrix} = \begin{bmatrix} \mathbf{k}_a & & & \\ & \mathbf{k}_{sz} & & \\ & & \mathbf{k}_{sy} & \\ & & & \mathbf{k}_t \end{bmatrix} \begin{bmatrix} \mathbf{q}_a \\ \mathbf{q}_{sz} \\ \mathbf{q}_{sy} \\ \mathbf{q}_t \end{bmatrix} \quad (4)$$

where $\mathbf{k}_a$ is the axial stiffness matrix, $\mathbf{k}_{sz}$ and $\mathbf{k}_{sy}$ are bending stiffness matrices about axes z and y, while $\mathbf{k}_t$ is the torsional stiffness matrix of the beam element. Elements of the stiffness matrix according to the second order theory may be obtained by solving the system of independent homogeneous differential equations (5) and by applying the corresponding boundary conditions of the clamped-clamped spatial beam element.

$$\begin{aligned} &EAu'' = 0 \\ &EI_z v^{IV} - Sv'' = 0 \\ &EI_y w^{IV} - Sw'' = 0 \\ &GI_x \varphi'' = 0 \end{aligned} \quad (5)$$

The first and the last equation of the system (5) define well known problems of the axial stress and torsion of the element according to the first order theory. By solving these equations, well known matrices of axial and torsional stiffnesses of the element are obtained, as can be found in any textbook related to the finite element method [1-3].

The second and the third equations of the system (5) define bending problems of an element subjected to the axial force S at beam ends, according to the second order theory, in

bending planes xy and xz, respectively. These equations may be presented as follows:

$$v^{IV} \pm k_z^2 v'' = 0 \tag{6}$$

$$w^{IV} \pm k_y^2 w'' = 0 \tag{7}$$

where

$$k_z = \sqrt{\frac{|S|}{EI_z}} \tag{8}$$

$$k_y = \sqrt{\frac{|S|}{EI_y}} \tag{9}$$

In equations (6) and (7), the plus sign corresponds to compressive axial force, while the minus sign corresponds to tension force. By finding an exact solution for these equations, the bending stiffness matrices of the compressed and/or tensioned beam element are obtained according to the second order theory, in planes xy and xz, $\mathbf{k}_{sz}$ i $\mathbf{k}_{sy}$. In case of compressive force, exact solutions for differential equations of the fourth order (6) and (7) are given in the form:

$$v(x) = a_1 + a_2 k_z x + a_3 \sin k_z x + a_4 \cos k_z x \tag{10}$$

$$w(x) = b_1 + b_2 k_y x + b_3 \sin k_y x + b_4 \cos k_y x \tag{11}$$

while in case of the tension force:

$$v(x) = \bar{a}_1 + \bar{a}_2 k_z x + \bar{a}_3 \operatorname{shk}_z x + \bar{a}_4 \operatorname{chk}_z x \tag{12}$$

$$w(x) = \bar{b}_1 + \bar{b}_2 k_y x + \bar{b}_3 \operatorname{shk}_y x + \bar{b}_4 \operatorname{chk}_y x \tag{13}$$

Unknown constants a_i, b_i, $\bar{a}$ and $\bar{b}$ are determined from the boundary conditions of the spatial beam element shown in Figure 1, which are:

$$\begin{aligned} &\text{čvor } i,\ x = 0\text{: } v(0) = v_i\ ;\ \varphi_z(0) = \varphi_{zi}\ ;\ w(0) = w_i\ ;\ \varphi_y(0) = \varphi_{yi} \\ &\text{čvor } j,\ x = L\text{: } v(L) = v_j\ ;\ \varphi_z(L) = \varphi_{zj}\ ;\ w(L) = w_j\ ;\ \varphi_y(L) = \varphi_{yj} \end{aligned} \tag{14}$$

In such a way, expression (10), i.e. displacement of an arbitrary point of an axis of an element in the plane xy, may be presented in matrix form as the generalized displacement function at element ends $\mathbf{q}_{sz}$ (derivation is the same as in linear theory):

$$v(x) = AC_{SZ}^{-1} q_{SZ} = N_{SZ} q_{SZ} \tag{15}$$

where:

$$\mathbf{q}_{sz}^T = \begin{bmatrix} v_i & \varphi_{zi} & v_j & \varphi_{zj} \end{bmatrix} = \begin{bmatrix} q_{sz}^1 & q_{sz}^2 & q_{sz}^3 & q_{sz}^4 \end{bmatrix} \tag{16}$$

$$N_{SZ} = [N_1(x)\ N_2(x)\ N_3(x)\ N_4(x)] \tag{17}$$

while:

$$\begin{aligned} N_1(x) &= \frac{1}{\Delta_z}[1-\cos\omega_z - \omega_z \sin\omega_z + \frac{x}{L}\omega_z \sin\omega_z - \sin\omega_z \sin k_z x + (1-\cos\omega_z)\cos k_z x] \\ N_2(x) &= \frac{1}{k_z \Delta_z}[\omega_z \cos\omega_z - \sin\omega_z + k_z(1-\cos\omega_z)x + (1-\cos\omega_z - \omega_z \sin\omega_z)\sin k_z x + \\ &\quad + (\sin\omega_z - \omega_z \cos\omega_z)\cos k_z x] \\ N_3(x) &= \frac{1}{\Delta_z}[1-\cos\omega_z + \frac{x}{L}\omega_z \sin\omega_z + \sin\omega_z \sin k_z x - (1-\cos\omega_z)\cos k_z x] \\ N_4(x) &= \frac{1}{k_z \Delta_z}[\sin\omega_z - \omega_z + \omega_z(1-\cos\omega_z)\frac{x}{L} - (1-\cos\omega_z)\sin k_z x + (\omega_z - \sin\omega_z)\cos k_z x] \end{aligned} \tag{18}$$

where:

$$\Delta_z = 2(1-\cos\omega_z) - \omega_z \sin\omega_z \tag{19}$$

$$\omega_z = L\sqrt{\frac{|S|}{EI_z}} \tag{20}$$

Expressions (18) represent interpolation functions of the bending problem in xy plane for a compressed beam element according to the second order theory. Their geometric interpretation is analogous to interpretation of the interpolation functions according to the first order theory. The interpolation function $N_i(x)$ (i=1,...,4) represents the deflection line of a clamped-clamped beam element subjected to axial forces S at element ends, due to the unit generalized displacement q_{sz}^i, while all other generalized displacements are equal to zero. By using expression (15), interpolation functions (18) and their derivatives with respect to *x*, the cross-sectional forces $T_y(x)$ and $M_z(x)$ may be obtained as:

$$\begin{aligned} T_y(x) &= -EI_z v'''(x) - Sv'(x) = (-EI_z N''' - SN'_{SZ}) \cdot q_{sz} \\ M_z(x) &= -EI_z v'''(x) = -EI_z N'_{SZ}\, q_{sz} \end{aligned} \tag{21}$$

By inserting into these expressions x = 0 and x = L, and taking care about the conventions of the positive signs of the generalized forces and cross-sectional forces, i.e. by using:

$$T_{yi} = -T_y(0)\ M_{zi} = M_z(0)\ T_{yj} = T_y(L)\ M_{zj} = -M_z(L) \tag{22}$$

matrix-form expressions for generalized forces at element ends are obtained for bending in xy plane as follows:

$$\mathbf{R}_{SZ} = \mathbf{k}_{SZ}\mathbf{q}_{SZ} \tag{23}$$

where:

$$k_{sz} = \frac{EI_z}{L^3\Delta_z}\begin{bmatrix} \omega_z^3 \sin\omega_z & L\omega_z^2(1-\cos\omega_z) & -\omega_z^3 \sin\omega_z & L\omega_z^2(1-\cos\omega_z) \\ L\omega_z^2(1-\cos\omega_z) & L^2\omega_z(\sin\omega_z - \omega_z\cos\omega_z) & -L\omega_z^2(1-\cos\omega_z) & L^2\omega_z(\omega_z - \sin\omega_z) \\ -\omega_z^3 \sin\omega_z & -L\omega_z^2(1-\cos\omega_z) & \omega_z^3 \sin\omega_z & -L\omega_z^2(1-\cos\omega_z) \\ L\omega_z^2(1-\cos\omega_z) & L^2\omega_z(\omega_z - \sin\omega_z) & -L\omega_z^2(1-\cos\omega_z) & L^2\omega_z(\sin\omega_z - \omega_z\cos\omega_z) \end{bmatrix} \tag{24}$$

Expression (24) represents the bending stiffness matrix of a compressed beam element in xy plane, according to the second order theory. Similarly, the bending stiffness matrix of a compressed beam element in xz plane can be obtained according to the second order theory:

$$k_{sy}=\frac{EI_y}{L^3\Delta_y}\begin{bmatrix} \omega_y^3\sin\omega_y & -L\omega_y^2(1-\cos\omega_y) & -\omega_y^3\sin\omega_y & -L\omega_y^2(1-\cos\omega_y) \\ -L\omega_y^2(1-\cos\omega_y) & L^2\omega_y(\sin\omega_y-\omega_y\cos\omega_y) & L\omega_y^2(1-\cos\omega_y) & L^2\omega_y(\omega_y-\sin\omega_y) \\ -\omega_y^3\sin\omega_y & L\omega_y^2(1-\cos\omega_y) & \omega_y^3\sin\omega_y & L\omega_y^2(1-\cos\omega_y) \\ -L\omega_y^2(1-\cos\omega_y) & L^2\omega_y(\omega_y-\sin\omega_y) & L\omega_y^2(1-\cos\omega_y) & L^2\omega_y(\sin\omega_y-\omega_y\cos\omega_y) \end{bmatrix} \quad (25)$$

where:

$$\Delta_y = 2(1-\cos\omega_y) - \omega y\sin\omega_y \quad (26)$$

$$\omega_y = L\sqrt{\frac{|S|}{EI_y}} \quad (27)$$

The bending stiffness matrix of the beam element subjected to the axial tension force may be derived in xy or xz plane according to the second order theory from differential equation of a beam under tension, given by (6) or (7), using the exact solution (12) or (13), by applying the same procedure as for the beam under compression. However, expressions for stiffness matrix of a compressed beam (24) or (25) can be used if ω_z, or ω_y, is substituted by $i\omega_z$, or $i\omega_y$, where i is the imaginary unit ($i^2=-1$), and also by applying the following relationships:

$$-i\sin i\omega_z = sh\omega_z \qquad \cos i\omega_z = ch\omega_z \quad (28)$$

$$-i\sin i\omega_y = sh\omega_y \qquad \cos i\omega_y = ch\omega_y \quad (29)$$

Therefore, the bending stiffness matrix of the beam element under tension, according to the second order theory, is given in plane xy or xz by expression (30) or (31):

$$k_{sz}=\frac{EI_z}{L^3\bar{\Delta}_z}\begin{bmatrix} \omega_z^3sh\omega_z & L\omega_z^2(ch\omega_z-1) & -\omega_z^3sh\omega_z & L\omega_z^2(ch\omega_z-1) \\ L\omega_z^2(ch\omega_z-1) & L^2\omega_z(-sh\omega_z+\omega_z ch\omega_z) & -L\omega_z^2(ch\omega_z-1) & L^2\omega_z(-\omega_z+sh\omega_z) \\ -\omega_z^3sh\omega_z & -L\omega_z^2(ch\omega_z-1) & \omega_z^3sh\omega_z & -L\omega_z^2(ch\omega_z-1) \\ L\omega_z^2(ch\omega_z-1) & L^2\omega_z(-\omega_z+sh\omega_z) & -L\omega_z^2(ch\omega_z-1) & L^2\omega_z(-sh\omega_z+\omega_z ch\omega_z) \end{bmatrix} \quad (30)$$

$$k_{sy}=\frac{EI_y}{L^3\bar{\Delta}_y}\begin{bmatrix} \omega_y^3sh\omega_y & -L\omega_y^2(ch\omega_y-1) & -\omega_y^3sh\omega_y & -L\omega_y^2(ch\omega_y-1) \\ -L\omega_y^2(ch\omega_y-1) & L^2\omega_y(-sh\omega_y+\omega_y ch\omega_y) & L\omega_y^2(ch\omega_y-1) & L^2\omega_y(-\omega_y+sh\omega_y) \\ -\omega_y^3sh\omega_y & L\omega_y^2(ch\omega_y-1) & \omega_y^3sh\omega_y & L\omega_y^2(ch\omega_y-1) \\ -L\omega_y^2(ch\omega_y-1) & L^2\omega_y(-\omega_y+sh\omega_y) & L\omega_y^2(ch\omega_y-1) & L^2\omega_y(-sh\omega_y+\omega_y ch\omega_y) \end{bmatrix} \quad (31)$$

where:

$$\bar{\Delta}_z = 2(1-ch\omega_z) + \omega_z sh\omega_z \quad (32)$$

$$\bar{\Delta}_y = 2(1-ch\omega_y) + \omega_y sh\omega_y \quad (33)$$

Now, complete stiffness matrices of the compressed or tensioned spatial beam element can be formed according to the second order theory, using stiffness matrices for separate stress states, i.e. $\mathbf{k}_a$, $\mathbf{k}_{sz}$, $\mathbf{k}_{sy}$ and $\mathbf{k}_t$. Elements of these stiffness matrices are placed at the corresponding positions of the complete stiffness matrix of the element, of the order 12, as determined by the assumed order of generalized displacement (1) and generalized forces (12), first for the node *i*, and then for the node *j*.

By introducing functions ϕ_i (i =1,...,8), stiffness matrices of the compressed and tensioned spatial beam element may be presented according to the second order theory in the same form, as given in (34). The functions ϕ_i (i =1,...,8) for cases of a compressed or tensioned beam element are given in Table 1, where D_z, ω_z, D_y, ω_y, $\bar{\Delta}_z$ and $\bar{\Delta}_y$ are given by expressions (19), (20), (26), (27), (32) and (33), respectively.

$$k=\begin{bmatrix} \frac{EA}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{EA}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{EI_z}{L^3}\varphi_1 & 0 & 0 & 0 & \frac{EI_z}{L^2}\varphi_2 & 0 & -\frac{EI_z}{L^3}\varphi_1 & 0 & 0 & 0 & \frac{EI_z}{L^2}\varphi_2 \\ 0 & 0 & \frac{EI_y}{L^3}\varphi_5 & 0 & -\frac{EI_y}{L^2}\varphi_6 & 0 & 0 & 0 & -\frac{EI_y}{L^3}\varphi_5 & 0 & -\frac{EI_y}{L^2}\varphi_6 & 0 \\ 0 & 0 & 0 & \frac{GI_x}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{GI_x}{L} & 0 & 0 \\ 0 & 0 & -\frac{EI_y}{L^2}\varphi_5 & 0 & \frac{EI_y}{L}\varphi_8 & 0 & 0 & 0 & \frac{EI_y}{L^2}\varphi_6 & 0 & \frac{EI_y}{L}\varphi_7 & 0 \\ 0 & \frac{EI_z}{L^2}\varphi_2 & 0 & 0 & 0 & \frac{EI_z}{L}\varphi_4 & 0 & -\frac{EI_z}{L^2}\varphi_2 & 0 & 0 & 0 & \frac{EI_z}{L}\varphi_3 \\ -\frac{EA}{L} & 0 & 0 & 0 & 0 & 0 & \frac{EA}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{EI_z}{L^3}\varphi_1 & 0 & 0 & 0 & -\frac{EI_z}{L^2}\varphi_2 & 0 & \frac{EI_z}{L^3}\varphi_1 & 0 & 0 & 0 & -\frac{EI_z}{L^2}\varphi_2 \\ 0 & 0 & -\frac{EI_y}{L^3}\varphi_5 & 0 & \frac{EI_y}{L^2}\varphi_6 & 0 & 0 & 0 & \frac{EI_y}{L^3}\varphi_5 & 0 & \frac{EI_y}{L^2}\varphi_6 & 0 \\ 0 & 0 & 0 & -\frac{GI_x}{L} & 0 & 0 & 0 & 0 & 0 & \frac{GI_x}{L} & 0 & 0 \\ 0 & 0 & -\frac{EI_y}{L^2}\varphi_6 & 0 & \frac{EI_y}{L}\varphi_7 & 0 & 0 & 0 & \frac{EI_y}{L^2}\varphi_6 & 0 & \frac{EI_y}{L}\varphi_8 & 0 \\ 0 & \frac{EI_z}{L^2}\varphi_2 & 0 & 0 & 0 & \frac{EI_z}{L}\varphi_3 & 0 & -\frac{EI_z}{L^2}\varphi_2 & 0 & 0 & 0 & \frac{EI_z}{L}\varphi_4 \end{bmatrix} \quad (34)$$

Table 1. Functions ϕ_i for a spatial beam element

Function	Axial force in spatial beam element	
	compression	tension
ϕ_1	$\frac{\omega_z^3\sin\omega_z}{\Delta_z}$	$\frac{\omega_z^3sh\omega_z}{\bar{\Delta}_z}$
ϕ_2	$\frac{\omega_z^2(1-\cos\omega_z)}{\Delta_z}$	$\frac{\omega_z^2(ch\omega_z-1)}{\bar{\Delta}_z}$
ϕ_3	$\frac{\omega_z(\omega_z-\sin\omega_z)}{\Delta_z}$	$\frac{\omega_z(sh\omega_z-\omega_z)}{\bar{\Delta}_z}$
ϕ_4	$\frac{\omega_z(\sin\omega_z-\omega_z\cos\omega_z)}{\Delta_z}$	$\frac{\omega_z(\omega_z ch\omega_z-sh\omega_z)}{\bar{\Delta}_z}$
ϕ_5	$\frac{\omega_y^3\sin\omega_y}{\Delta_y}$	$\frac{\omega_y^3sh\omega_y}{\bar{\Delta}_y}$
ϕ_6	$\frac{\omega_y^2(1-\cos\omega_y)}{\Delta_y}$	$\frac{\omega_y^2(ch\omega_y-1)}{\bar{\Delta}_y}$
ϕ_7	$\frac{\omega_y(\omega_y-\sin\omega_y)}{\Delta_y}$	$\frac{\omega_y(sh\omega_y-\omega_y)}{\bar{\Delta}_y}$
ϕ_8	$\frac{\omega_y(\sin\omega_y-\omega_y\cos\omega_y)}{\Delta_y}$	$\frac{\omega_y(\omega_y ch\omega_y-sh\omega_y)}{\bar{\Delta}_y}$

As opposed to the stiffness matrix according to the first order theory [1] where all elements are constants and depend on the geometry and mechanical properties of the beam element, the elements of the stiffness matrix according to the second order theory (34) are trigonometric or hyperbolic functions, depending on whether the beam element is compressed or tensioned, which is related to the parameters ω_z and ω_y, i.e. on the axial force S.

By reaching an approximate solution of differential equations due to the second order theory (6) and (7), the stiffness matrix of the second order theory is obtained as the sum of two matrices, the linear stiffness matrix (according to the first order theory) and the geometric stiffness matrix, see [1]. This second approach involving geometric stiffness matrix is more suitable (e.g. for programming), which is why it is generally implemented in commercial programs for structural analysis.

2.2. Equivalent load vector

The basic equation for an unloaded beam finite element is given by expression (3). If a load is distributed along the beam element axis, the basic equation for such beam finite element is:

$$\mathbf{R} = \mathbf{kq} - \mathbf{Q} \tag{35}$$

where:

$$\mathbf{Q}^T = [N_i^Q T_{yi}^Q T_{zi}^Q M_{xi}^Q M_{yi}^Q M_{zi}^Q N_j^Q T_{yj}^Q T_{zj}^Q M_{xj}^Q M_{yj}^Q M_{zj}^Q] \tag{36}$$

is the equivalent load vector, i.e. the concentrated load at the ends of the beam element. Such concentrated forces at element ends are replacing the external load distributed along the element axis. The same positive sign convention as for the generalized forces is assumed for the elements of the equivalent load vectors, Figure 2.

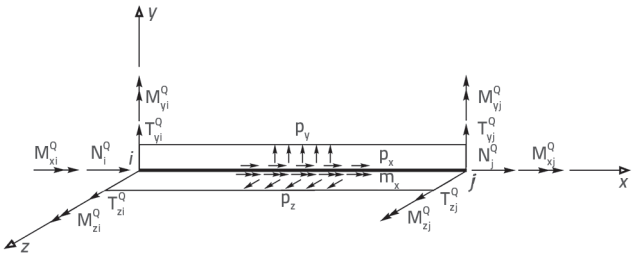


Figure 2. Equivalent load at beam element nodes

The equivalent load vector of the spatial beam finite element **Q** can be obtained in the same way as the stiffness matrix, by separating the spatial state of stress into: axial load, bending in xy plane, bending in xz plane, and torsion.

Elements of the equivalent load vector due to axial load distribution and torsional moment distribution along the element axis according to the second order theory, are the same as those based on the first order theory, since differential equations in case of axial and torsional loading for the second order theory are the same as those for the first order theory. This is not the case with elements of the equivalent load vector due to transverse load distribution. They can be obtained by solving the following differential equations:

$$EI_z v^{IV} - Sv'' = p_y \tag{37}$$

$$EI_y w^{IV} - Sw'' = p_z \tag{38}$$

and by applying the corresponding boundary conditions of the clamped-clamped beam element.

Equations (37) and (38) define the bending problems of an element subjected to transverse load p_y and p_z and to axial force S at element ends, in planes xy and xz, according to the second order theory. These equations can be presented as follows:

$$v^{IV} \pm k_z^2 v'' = \frac{p_y}{EI_z} \tag{39}$$

$$w^{IV} \pm k_y^2 w'' = \frac{p_z}{EI_y} \tag{40}$$

where k_z and k_y are given in expressions (8) and (9).

In equations (39) and (40), the plus sign corresponds to the axial compressive force, while the minus sign corresponds to the axial tension force. By resolving these equations using the method of initial parameters [4], equivalent load vectors of the compressed and/or tensioned beam element are obtained in the xy and xz planes, according to the second order theory, $\mathbf{Q}_{sz}$ and $\mathbf{Q}_{sy}$.

$$\mathbf{Q}_{sz}^T = \frac{p_y L}{2}\left[1 \;\; \frac{L}{6}\gamma_z \;\;\; 1 \;\; -\frac{L}{6}\gamma_z\right] \tag{41}$$

$$\mathbf{Q}_{sy}^T = \frac{p_z L}{2}\left[1 \;\; -\frac{L}{6}\gamma_y \;\;\; 1 \;\; \frac{L}{6}\gamma_y\right] \tag{42}$$

The functions g_y and g_z, representing the influence of the second order theory, are given in Table 2 for cases of the compressed or tensioned spatial beam element, where ω_z and ω_y are given in expressions (20) and (27).

It is now possible to form the complete equivalent load vector of the compressed and tensioned beam element in 3D space according to the second order theory, using equivalent load vectors for separate stress states $\mathbf{Q}_a$, $\mathbf{Q}_{sz}$, $\mathbf{Q}_{sy}$ and $\mathbf{Q}_t$ by placing their elements into the corresponding positions determined in expression (36). By introducing functions g_y and g_z (Table 2), equivalent load vectors of the compressed and tensioned beam element in space, according to the second order theory, may be written as given in (43).

$$\mathbf{Q}^T = \frac{L}{2}\left[p_x p_y p_z m_x - \frac{p_z L}{6}\gamma_y \frac{p_y L}{6}\gamma_z p_x p_y p_z m_x \frac{p_z L}{6}\gamma_y - \frac{p_y L}{6}\gamma_z\right] \tag{43}$$

Table 2. Functions γ_y and γ_z for spatial beam element

Function	Axial force in spatial beam element	
	compression	tension
γ_y	$\gamma_y = -\frac{6\omega_y \sin\omega_y + 12(\cos\omega_y - 1)}{\omega_y^2(1-\cos\omega_y)}$	$\gamma_y = -\frac{6\omega_y \mathrm{sh}\omega_y + 12(1-\mathrm{ch}\omega_y)}{\omega_y^2(1-\mathrm{ch}\omega_y)}$
γ_z	$\gamma_z = -\frac{6\omega_z \sin\omega_z + 12(\cos\omega_z - 1)}{\omega_z^2(1-\cos\omega_z)}$	$\gamma_z = -\frac{6\omega_z \mathrm{sh}\omega_z + 12(1-\mathrm{ch}\omega_z)}{\omega_z^2(1-\mathrm{ch}\omega_z)}$

Therefore, in the equivalent load vector of the beam element according to the second order theory, the concentrated bending moments at element ends are function of the axial force in element, and only they are different from bending moments according to the first order theory, while the other components are equal to the corresponding components according to the first order theory.

2.3. Reduction of stiffness matrix and equivalent load vector

If some constraints at finite element ends are removed, i.e. if one or more nodal generalized forces are equal to zero, then the stiffness matrix and the equivalent load vector have to be reduced.

Components of the vector of generalized forces R at element ends, where the k^{th} element is equal to zero, having in mind Eq. (35), may be written as follows:

$$R_i = \sum_{j=1}^{n} k_{ij}^r q_j - Q_i^r \quad (i = 1, 2, ..., n) \tag{44}$$

where:

$$k_{ij}^r = k_{ij} - k_{ik}\frac{k_{kj}}{k_{kk}} \quad (i = 1, 2, ..., n \quad j = 1, 2, ..., n) \tag{45}$$

$$Q_i^r = Q_i - \frac{k_{ik}}{k_{kk}} Q_k \quad (i = 1, 2, ..., n) \tag{46}$$

In expressions (44)-(46), the notation *r* in the exponent denotes that it corresponds to the reduced quantity. The order of the reduced stiffness matrix remains n, i.e. 12, but the elements of the k^{th} row and k^{th} column are equal to zero. If there are more generalized forces at element ends that are equal to zero, this reduction procedure should be repeated, but starting with the already reduced stiffness matrix and reduced equivalent load vector resulting from previous reduction. The expression for the vector of generalized forces at element ends is now given as:

$$\mathbf{R} = \mathbf{k}^r\mathbf{q} - \mathbf{Q}^r \tag{47}$$

where $\mathbf{k}^r$ is the reduced stiffness matrix, $\mathbf{q}$ is the vector of generalized displacements, and $\mathbf{Q}^r$ is the reduced equivalent load vector, containing all releases at element ends.

2.4. Transformation matrix

The above analysis of the beam element in space was carried out in the local coordinate system *ixyz*, defined for the beam element. Before analysis of the structure, i.e. the system of mutually connected elements, it is necessary to transform all vectors and matrices from the local coordinate system *ixyz*, which applies individually to each beam element, to the global system OXYZ. In order to establish relationships between the axes of the local and global coordinate systems, the position of each beam element with respect to the global system must be defined.

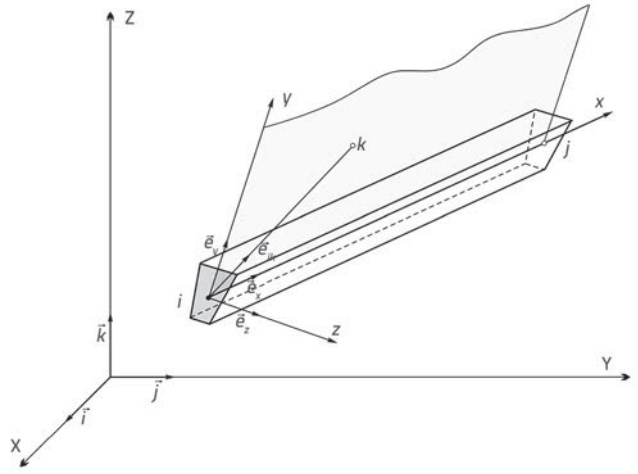


Figure 3. Local and global coordinate system of spatial beam element

In order to define spatial position of a beam element, the position of nodes *i* and *j* (element ends) must be known, as well as the position of point *k*, Figure 3. The point *k* is used to define spatial position of the principal central axes of the element's cross section, and may be chosen arbitrarily in space, the only condition being that it belongs to the plane defined by the axis *x* of the element and one of the central principal axes of the cross section, say, axis *y*.

Now that the position in space of the beam element has been defined, the relationships between the vector of generalized displacements and generalized forces in the local and global system, where the global quantities are denoted by *, may be presented as:

$$\mathbf{q} = \mathbf{T}\mathbf{q}^* \tag{48}$$

$$\mathbf{R} = \mathbf{T}\mathbf{R}^* \tag{49}$$

where:

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{\Lambda} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Lambda} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\Lambda} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{\Lambda} \end{bmatrix} \tag{50}$$

is the matrix for transformation of quantities from the global into the local coordinate system, which is presented by rotation matrix as submatrices:

$$\boldsymbol{\Lambda} = \begin{bmatrix} \cos(x,\mathrm{X}) & \cos(x,\mathrm{Y}) & \cos(x,\mathrm{Z}) \\ \cos(y,\mathrm{X}) & \cos(y,\mathrm{Y}) & \cos(y,\mathrm{Z}) \\ \cos(z,\mathrm{X}) & \cos(z,\mathrm{Y}) & \cos(z,\mathrm{Z}) \end{bmatrix} \tag{51}$$

In order to obtain the stiffness matrix of the beam element in the global system, it is necessary to insert expressions (48) and

(49) into Eq. (3), and pre-multiply the obtained relation by $\mathbf{T}^T$. In such a way, the following relation is obtained:

$$\mathbf{R}^* = \mathbf{k}^*\mathbf{q}^* \tag{52}$$

where:

$$\mathbf{k}^* = \mathbf{T}^T\mathbf{k}\mathbf{T} \tag{53}$$

is the stiffness matrix of the beam element in the global coordinate system.

3. Analysis of system of elements according to second order theory

In the above analysis, beam elements are treated as independent parts of the structure. However, when analysing the structure, i.e. the system of mutually connected elements, one must take into account their connectivity. Elements of the system, at the node where they are connected, must satisfy the compatibility conditions of displacements and also the force equilibrium conditions. According to the second order theory, equilibrium conditions at the nodes of the system can be presented as the matrix equation:

$$\mathbf{K}^*\mathbf{q}^* = \mathbf{S}^* \tag{54}$$

where: $\mathbf{K}^*$ is the stiffness matrix of the system of elements according to the second order theory, $\mathbf{q}^*$ is the vector of generalized displacements of the nodes of the system, and $\mathbf{S}^*$ is the loading vector according to the second order theory, which represents the sum of the vector of a given external forces acting at nodes of the system and the equivalent load vector of the system according to the second order theory.

Since the stiffness matrix is singular, the boundary conditions, i.e. conditions of support of the system of elements, must be inserted in order to solve the system of equations (54). One way to insert the boundary conditions is to add relatively large constants at the diagonal of the stiffness matrix of the system at positions corresponding to restrained degrees of freedom. This is equivalent to adding a large stiffness to the system at the position and in the direction of the restrained generalized displacement.

When system node displacements according to the second order theory are determined by solving equations (54), where the boundary conditions were previously inserted, the generalized forces at the ends of individual beam elements are determined, based on the second order theory, in the global coordinate system from equations:

$$\mathbf{R}_k^* = \mathbf{k}_k^*\mathbf{q}_k^* - \mathbf{Q}_k^* \qquad (k = 1, 2, ..., M) \tag{55}$$

where $\mathbf{k}_k^*$ and $\mathbf{Q}_k^*$ are the stiffness matrix and the equivalent load vector of element k according to theory of the second order in the global coordinate system, while M is the total number of beam elements in the system.

Generalized forces at the ends of individual elements according to theory of second order, in the local coordinate system, are determined from relation:

$$\mathbf{R}_k = \mathbf{T}_k\mathbf{R}_k^* \qquad (k = 1, 2, ..., M) \tag{56}$$

where $\mathbf{T}_k$ is the transformation matrix of element k from the global into the local coordinate system.

Calculation according to the second order theory is performed in this paper in such a way that the normal forces in the first iteration are assumed as having been obtained from the previous calculation according to the first order theory. In the following iterations, normal forces are assumed to have been obtained from the previous iteration according to the second order theory, and the iterative procedure is repeated until the differences between displacements obtained in two successive iterations become less than some pre-assumed threshold value, or the calculation is stopped after certain number of iterative cycles.

4. Computer code ALIN

Based on preceding discussion, an appropriate computer code called ALIN (**A**naliza **LI**nijskih **N**osača, i.e. Analysis of Framework Structures) [5] was developed as a tool for the analysis of the spatial and planar frame and truss structures. The ALIN code has been designed for static analyses (according to the first and second order theories), stability analyses (determination of critical load), and dynamic analyses (eigenvalue problems and direct numerical integration of differential equations of motion). The program is written in the programming language C++ [6-9]. It is currently a console application based on the input file of XML type, while the output results are presented in the corresponding output text files. Just like other similar programs, there are three basic program modules that make a single entity. Tush, there are modules for:

- input data, i.e. for definition of a problem
- formation and solution of appropriate equations
- processing and presentation of results.

Therefore, the user uses the input file to define the structure and the problem to be solved by providing the data about the structure, load and the type of analysis to be used by the program. The validity of input data is checked through analysis of input file. If error is found in input data (e.g. wrong input number of the node, material, cross-section or element not present in the input list), the program automatically stops and submits information about the error it has detected. Based on input data, the program calculates effects in the structure and presents them through output text files.

4.1. Input and output files

Input data are given in form of XML files, using the DOM (Document Object Model) concept and C++ library TinyXML [10].

This paper presents only the organization of the input file of XML type, while all input data needed to describe the structure and the problem are given in [5].

The input file is organised as follows.
The basic, or root, element of the input file Some_Name.xml is <ALIN>...</ALIN>. The root element contains the following blocks:

- <System>...</System>, where the general problem data are given,
- <Property>...</Property>, where data about materials and cross-sections are given,
- <Nodes>...</Nodes>, where data about nodal points, degrees of freedom, and boundary conditions are given,
- <Elements>...</Elements>, where data about finite elements are given,
- <Loading>...</Loading>, where loading data are given.

Output files consist of text files. Their names depend on output results (output values), i.e. on the type of analysis. There is also an output log file for control display of input data, whose name is given in the system info part (block <OutFile>).
If the type of analysis is STATIC, i.e. if calculation is made according to the first order theory, output results are cross-sectional forces at end points of each element in local coordinates (NodalForces.txt) and also in the global coordinate system (GlobalNodalForces.txt), as well as displacements of all nodes of the system in global coordinates (NodalDisplacements.txt).
If the type of analysis is STABILITY, i.e. if calculation is conducted according to the second order theory, or if the aim is to determine the critical buckling load, output results are cross-sectional forces at ends of all elements, in local and global coordinate systems, as well as displacements of all nodal points of the system in global coordinates, determined according to the second order theory, or the value of the critical buckling parameter. The corresponding output files are formed in the module DYNAMIC, which is not considered in the scope of this analysis.

4.2. Organization of ALIN

The basic class of the ALIN program is the class called **Model** that describes the model of the structure. It contains the following main classes, i.e. their instances: **System**, **Materials**, **Section**, **Node_List**, **Elements**, **BasicLoads**, **Loads**, **LoadAdd**, **TimeForce** i **GlobalEqs** (Figure 4). In addition, private members of the class **Model** are the names of the input xml file and the output control file, as well as the corresponding control indicators that check whether all necessary input data have been provided, etc.

The first task of the class **Model** is collection of input data and control output of read input into the output control file. Input data are obtained by reading the input xml file and it is done in the member function of the class Model::inputXML().Local variables of the corresponding type are defined within this function in order to receive input data through xml-parsing (according to TinyXML approach). After reading data from a certain input block and transferring them to local variables, which exist only to obtain data after parsing, the data are transferred to variables of the same corresponding type, which are private members of the class **Model**.
The next tasks of the class **Model** are to form the model of the structure, to obtain solution of the corresponding equations of the problem (statics, stability, dynamics), and to present the results. These tasks depend on the type of analysis and are performed within the corresponding member functions of the class **Model**, which are not considered in this analysis. The C++ library **SMV_lib** is used for efficient manipulations with vectors and matrices, as well as for solving equations of the problem at hand, [6].

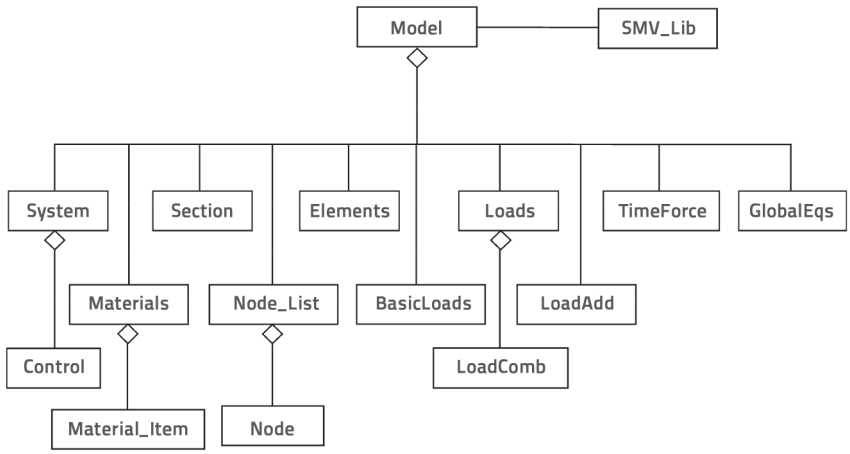


Figure 4. Object model of the class Model with basic classes

Figure 4 presents the object model of the class **Model** (the basic class of the program ALIN), i.e. it presents the basic relations between classes, which are briefly discussed in the following text.

4.2.1. Class System

General data about the considered problem are defined within the class **System**. Basic private members of the class are the vector of strings with titles of the problem (vector<string>Titles) and instance of the class **Control** (**Control** control), which contains all control information about the problem consisting of a series of ENUM variables and a series of INT variables (control numbers). Part of the file Enums.h is presented below, with some of the variables that control the flow of the program. ENUM controls are:

```
//-------- general
enum SPACE {s2D = 2, s3D = 3};
```

```
enum ANALYSIS {STATIC, DYNAMIC, STABILITY};
//-------- stability
enum STAB_ANAL {SECOND, CRITICAL};
//--------- structure and finite elements
enum STRUCTURE {TRUSS, FRAME, TWBEAM, CABLE, MIXED};
enum ELEMENTS {TRUSS_ELE, BEAM_ELE, TWBEAM_ELE, CABLE_
ELE };
//--------- material and cross sections
enum MATERIAL {CONCRETE, STEEL, OTHER};
enum SECTION {FULL, THIN_WALLED};
enum FULL_SECTION {RECT, CIRC, T_SEC, I_SEC, GEN_SEC, MIXED_
SEC};
```

INT controls are:

No_Elem	(total number of finite elements)
No_Truss	(number of truss elements)
No_Beams	(number of beam elements)
No_TWBeams	(number of thin-walled elements)
No_Cables	(number of cable elements)
No_Sect	(total number of different cross-sections)
No_TWSect	(total number of different thin-walled cross-sections)
No_Mat	(total number of different materials)
No_LC	(total number of load cases)
No_Eqs	(total number of degrees of freedom of finite elements)

4.2.2. Class Materials

Class **Materials** contains data about different materials that can be used in the analysis. Private members of this class are the vector with pointers to different materials, i.e. pointers to the class **Material_Item** (vector<Material_Item*>Mat_List), as well as the bool indicator, if there is such a vector.
Class **Material_Item** contains all data about materials, such as the type of material, its reference number and name, as well as its properties: modulus of elasticity, Poisson's coefficient, shear modulus, unit weight, and mass density.

4.2.3. Class Section

Class **Section** (Figure 5) contains the data about various cross-sections. Private members of this class are vectors with pointers to various types of cross-sections, i.e. pointers to the class **Rectangle** (for the fully rectangular cross-section) vector<Rectangle*>Rect, pointer to the class **Circle** (for circular section) vector<Circle*>Circ, and pointer to the class **G_Section** (for general cross-section) vector<G_Section*>GSec, as well as the corresponding bool indicators.
When using the finite element method, the shape of the cross-section is irrelevant, i.e. only its properties, i.e. the area and corresponding moments of inertia, are relevant. Consequently, the class **Property** is implemented in the class **Section** in order to transfer to finite elements all relevant properties, i.e. numerical values for each cross-section. That class is the same as the class related to the general cross-section **G_Section** (where pre-calculated values are given by the user).
Basic private members of classes **Rectangle** and **Circle** are the width and height of the rectangular cross-section, or diameter of the circular section, for which the area and moments of inertia of the cross-section are calculated. The remaining private members of all three classes **Rectangle**, **Circle** and **G_Section** are: unique identification number (unique within its class, i.e. within the group of sections of the same shape), name of cross-section, info about the space (whether the problem is related to a plane (2D) or space (3D)), area of section (A) and moments of inertia (I_1, I_2 and I_3), as well as the corresponding bool indicators.

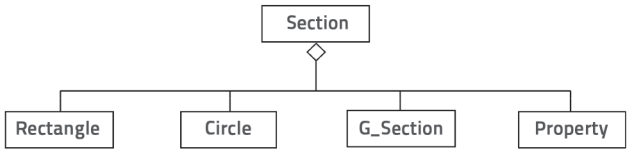


Figure 5. Object model of cross section (class Section)

4.2.4. Class Node_List

Class **Node_List** is the collection of all nodal points in the numerical model. Private members of this class are the vector with pointers to all nodal points, i.e. pointers to class **Node** (vector<Node*>Nodes), as well as the bool indicator if the vector exists. Private members of this class are instances of classes **Joint**, **DOFs** and **Boundary**, which are used to define data about nodal points, degrees of freedom, and boundary conditions. Also, private members of the class are the identification number of the node, its name, and also info about whether the problem is planar or in 3D space, which is transferred through the instance of the class **Joint**. Class Node is presented in Figure 6.

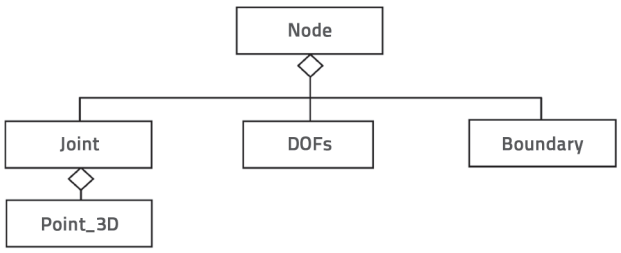


Figure 6. Object model of class Node

Class **Joint** is used to manipulate nodal points of the numerical model, so its basic private member is the instance of the class **Point_3D**. Other private members of class **Joint** are the identification number of the joint, name of the joint, and info about the space that is directly transferred from the class **Point_3D**.
Representation of points in plane (2D) or in space (3D), and manipulation of points, is performed using the class **Point_3D**. Private members of this class are three coordinates, X, Y and Z, as well as the info about the space, variable Space, i.e. the spatial indicator 2D or 3D. If the problem is 2D, it is assumed

that it is in the plane XY, and so the coordinate Z is Z=0. In the case when only X and Y coordinates are provided as input, it is automatically detected that it is a 2D problem. However, if a Z coordinate is also provided, then it is a 3D problem. This must be in accordance with the basic Space variable from class Control. Elementary functions related to manipulation of points, such as addition, subtraction, scaling, comparison, etc., may be performed within the class **Point_3D**.

Class **DOFs** is used to manipulate data related to degrees of freedom at nodal points of the numerical model. Private members of class **DOFs** are a vector with 6 bool elements: bool DOF[6] and info about the space, variable Space. Elements of vector DOF[i] are related to displacements in the direction of X, Y and Z axes, and rotations about X, Y and Z axes, i.e. UX, UY, UZ, RX, RY and RZ. If DOF[i]=false then the degree of freedom number "i" does not exist while, if it is true, then the degree of freedom "i" is defined. Initially, in constructor, all elements are set to false. Degrees of freedom are automatically set depending on whether the problem is 2D or 3D: for Space=s2D three degrees of freedom are defined: UX, UY and RZ, so DOF=[1,1,0,0,0,1], while for s3D, DOF=[1,1,1,1,1,1]. Therefore, the data related to degrees of freedom are automatically assigned, except in the case of external or internal restraints (e.g. internal hinge) when they are additionally allocated.

Class **Boundary** is used for manipulation of data related to boundary conditions at nodal points of the numerical model. The basic private member of this class is the vector of 6 bool elements: bool CON[6]. Elements of this vector represent degrees of freedom, UX, UY, UZ, RX, RY and RZ. Initially, in constructor, all elements are set to false, and so all generalized displacements are permitted. If CON[i] = true, then the degree of freedom "i" is restrained, i.e. the corresponding generalized displacement does nor exist and is set to zero.

4.2.5. Class Elements

Class **Elements** (Figure 7) involves storing data about various types of finite elements in plane or in space.

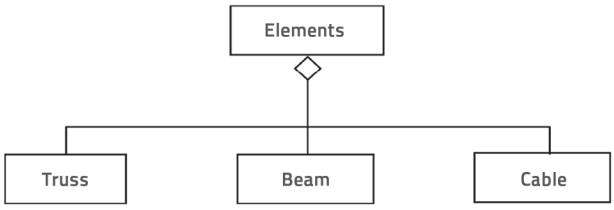


Figure 7. Object model of class Elements

Basic private members of class **Elements** are vectors with pointers to various types of finite elements (vector<Truss*> Truss_Ele, vector<Beam*> Beam_Ele, vector<Cable*> Cable_Ele), as well as the corresponding bool indicators whether these vectors exist or not. At this moment, the implemented elements are: truss (class **Truss**), beam (class **Beam**) and cable (class **Cable**) finite elements. Class **TWBeam** is also partly implemented, but only classes **Truss** and **Beam** are presented here**,** since the class **Cable** is not used in this paper, while the class **TWBeam** is not completely functional**,** and is not used in this analysis.

Class **Truss** represents the class whose members are planar or spatial truss beams. Finite elements of type Truss are defined by its identification number, which is unique within the class Truss, and by nodal points I and J, as well by data about the material and cross section. It is possible, as an option, to assign a name to each element (or to some elements). This class is responsible for the transformation matrix, for the stiffness matrix (local and global) and for the equivalent load vector. Private members of the class Truss are the vectors of the local and global displacement of nodal points I and J, i.e. of end points of the truss element.

Class **Beam** is a class whose members are planar or spatial beam elements. Elements of Beam type are defined by its identification number (unique for each element within the Beam group), nodal points I and J (also the node K for the spatial beam element, which is used to define the position of its local coordinate axes), and also data related to material and cross section. Of course, a name can be assigned to each beam element. To be able to assign member releases at beam ends, private members of the **Beam** class are two vectors of 6 bool elements, each related to releases of degrees of freedom at nodes I and J, as well as two corresponding bool indicators which indicate whether releases of degrees of freedom at nodes exist or not. Initially, in constructor, these elements are set to false. Class **Beam** is responsible for the transformation matrix, local and global stiffness matrices, and equivalent load vector. Private members of this class are also vectors of local and global displacements of nodal points of beam elements.

4.2.6. Class BasicLoads

Class **BasicLoads** (Figure 8) is the collector for all basic load cases, and so its basic private member is the vector with pointers to class **LoadCase** (vector<LoadCase*> LCases). Other private members of the class are the vector with load case names (vector<string> LCNames), as well as the corresponding bool indicators.

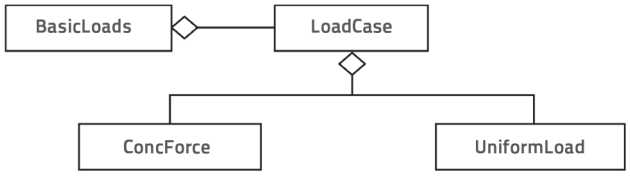


Figure 8. Object model of class BasicLoads

Class **LoadCase** represents the class that contains all loads that contribute to a basic load case. Private members of this class are vectors with pointers to classes **ConcForce** (for concentrated loads at nodal points) and **UniformLoad** (for loads distributed along elements). Therefore, vector<ConcForce*>Forces and vector<UniformLoad*>Uniform are members of the class.

In addition, private members of this class are the name and reference number of the load case (string Name and int Number), as well as the corresponding bool indicators. It should be mentioned that a single basic load case may contain an arbitrary number of concentrated and uniformly distributed loads.

Class **ConcForce** is the class that represents concentrated forces at one node. Therefore, private members of this class are a pointer to the nodal point, i. e. to class **Node** (Node* N), as well as a vector double Force[6] with six elements that represent components of the concentrated force (FX, FY and FZ) and components of the concentrated couple (MX, MY and MZ). These components are defined with respect to the global coordinate system. Private members of this class are a vector bool DIR[6] with elements equal to 0 or 1 depending on whether the corresponding component of the concentrated load exists or not, and also the corresponding bool indicators.

Class **UniformLoad** represents loads that are uniformly distributed along elements of Truss and Beam type. Basic private members of this class are pointers to classes Truss and Beam (Truss*TrussEle and Beam*BeamEle), along which the load is defined, as well as the vector double Uniform [6], whose elements are components of distributed forces px, py, pz, and components of distributed couples mx, my i mz. These components are defined with respect to the local coordinate systems of the studied elements. Components of distributed forces px, py, pz are used for elements of Truss type only. Just like for the class ConcForce, the bool vector DIR[6] is defined, and its elements have the values of 0 or 1 depending on whether the corresponding component of distributed forces and distributed couples exists or not. Consequently, private members of the class UniformLoad are the vector bool DIR [6] and the corresponding bool indicators.

Class **Loads** is only the collector for all load combinations. Members of this class are the vector with pointers to the class **LoadComb**, i.e. to load combinations (vector<LoadComb*>Load), as well as the bool indicator if at least one load combination is defined or not.

Class **LoadComb** represents one combination of basic load cases. Their private members are the name of the load combination (string LCName), vector with multipliers of the basic load cases (vector<double>LCFactor), as well as the corresponding bool indicators. The vector with multipliers of the basic load cases is initialized to have as many elements as the number of defined basic load cases, with all factors initialized to 0.0. It is assumed that the first load factor corresponds to the first load case, the second factor to the second load case, etc. When preparing the input data, only the non-zero load factors are given, together with the number of the corresponding load case.

Classes **LoadAdd** and **TimeForce**, presented in Figure 4, are beyond the scope of this paper. Class **LoadAdd** contains all loads that act after the initial equilibrium configuration due to permanent load of a cable-stayed bridge is established, while the class **TimeForce** contains all dynamic loads acting upon a frame structure or upon a cable-stayed bridge, [5].

4.2.8. Class GlobalEqs

Class **GlobalEqs** contains the global stiffness matrix and the global loading vector. The class is responsible for their formation and manipulation, i.e. for resolving the corresponding equations of the problem. Its basic private members are the stiffness matrix (Matrix* K), the loading vector (Vector* Load) and displacement vector (Vector* Disp), as well as the corresponding bool indicators.

4.2.9. Matrix library SMV_lib

Matrix library **SMV_lib** is used for manipulations of matrices, as well as for resolving the corresponding equations of the problem. The following main classes of the library are used:

- class **Arrays,** used for manipulations of vectors and matrices,
- class **LinEqs,** used for solving linear algebraic equations,
- class **EigVal**, used for solving the eigenvalue problem.

Class **LinEqs** contains classes **Cholesky** and **Crout**, which are used for solving the system of linear algebraic equations by LU decomposition. The method is automatically chosen depending on the structure of the coefficient matrix: Cholesky for symmetric, and Crout for nonsymmetric matrices. Class **EigVal** contains classes **Jacobi** and **Lanczos**, which are used for solving the eigenvalue problem. More detailed information about the C++ matrix library SMV_lib is given in [6].

5. Numerical examples

The analysis of two planar frames and two spatial frames according to theory of the first and the second order, due to given loading, is performed in this paper using the program ALIN. The results obtained are compared with results acquired using the program TOWER, and the influence of transverse forces on deformation is neglected. Calculation according to the second order theory, using the program TOWER, was performed by discretization of beams into smaller finite elements in order to obtain more accurate results [11]. Using the program ALIN, the same discretization was kept as in the case of the first order theory, namely one beam was one finite element.

5.1. Example 1

Figure 9 represents planar frames subjected to uniformly distributed load along axis of the beam 1-2, and to the action of concentrated vertical forces at nodes 2 and 3. The first frame is subjected to compressive force at node 2, and the second to tension force, while at node 3 both frames are subjected to compressive force. This example is selected to check the program ALIN with respect to the second order theory since the beam 1-2, which is subjected to uniformly distributed load, is under compression in the first frame, while it is under tension in the second frame. Consequently, there is a difference in stiffness matrices, and also in equivalent load vectors. In order to obtain

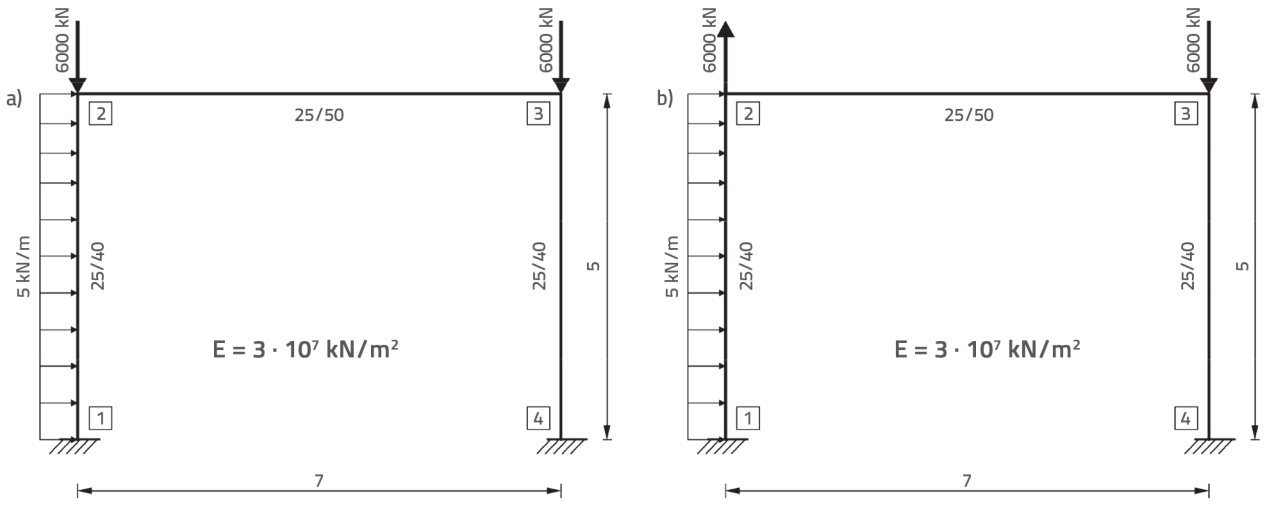


Figure 9. Planar frames. Geometry and load: a) The first frame; b) The second frame

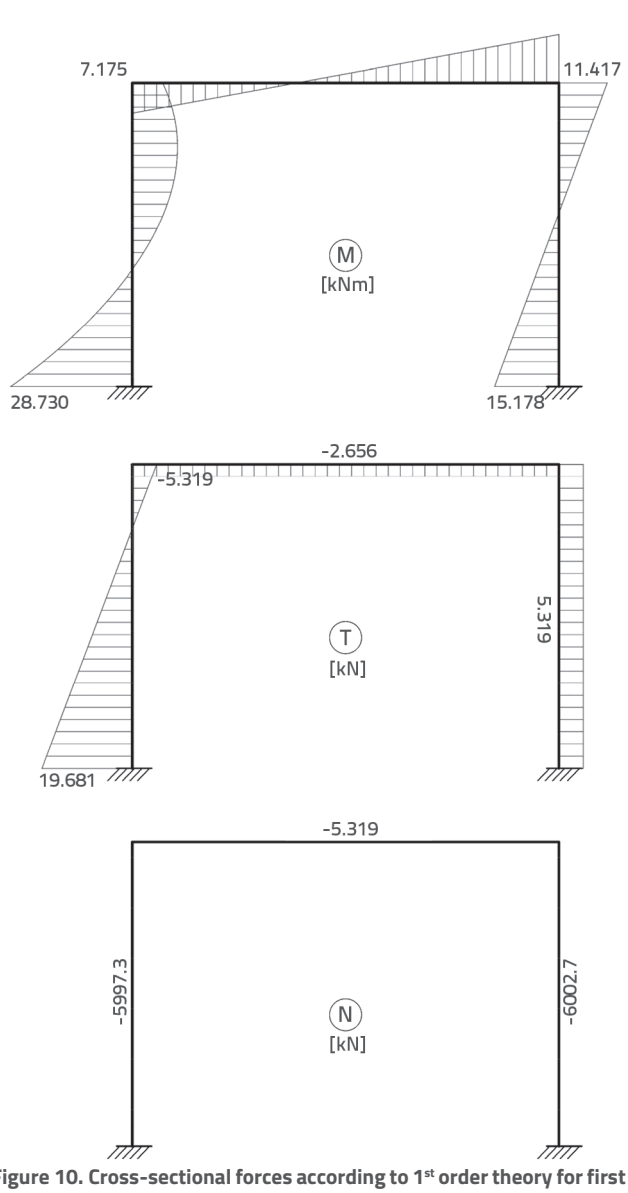


Figure 10. Cross-sectional forces according to 1st order theory for first frame

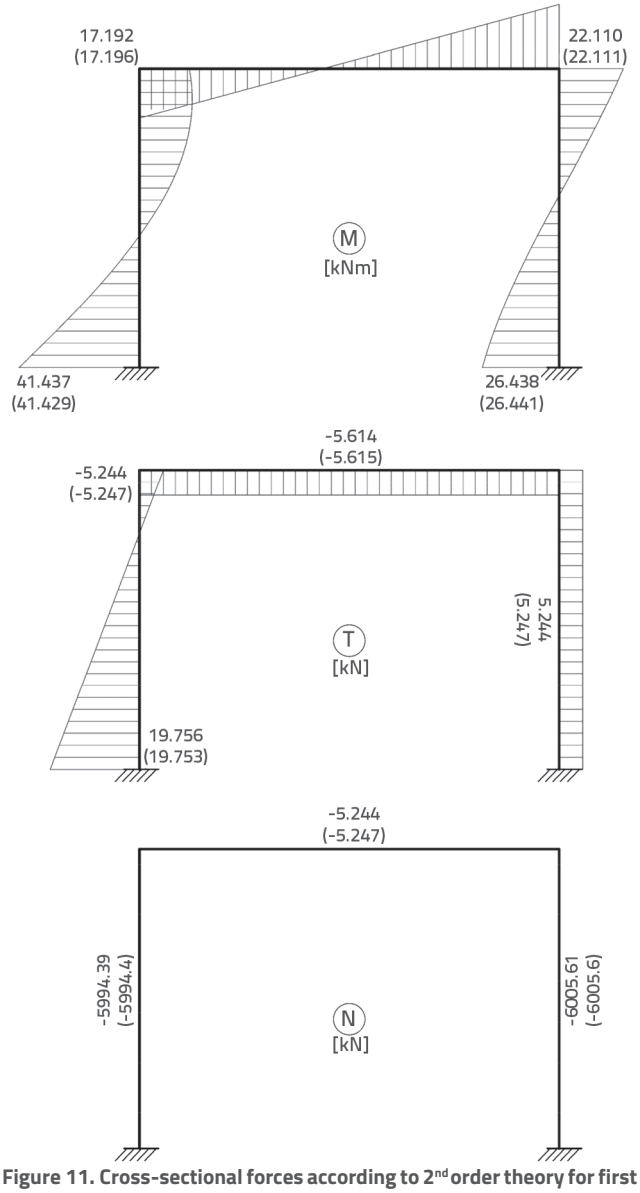
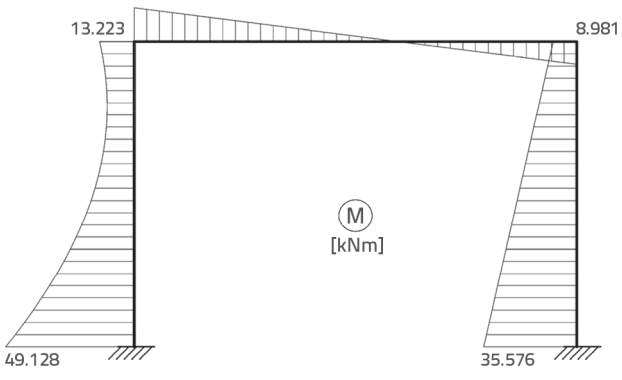
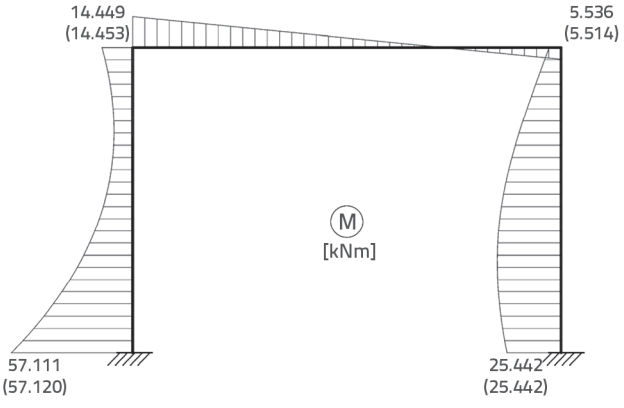
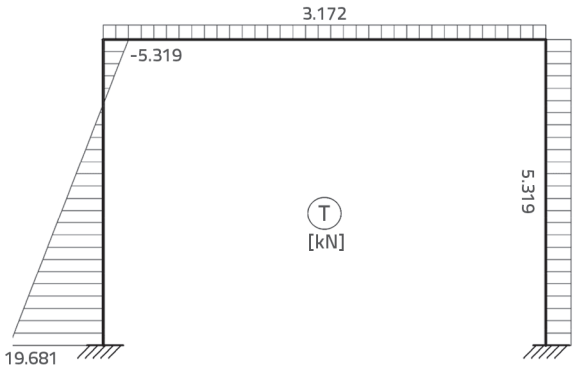
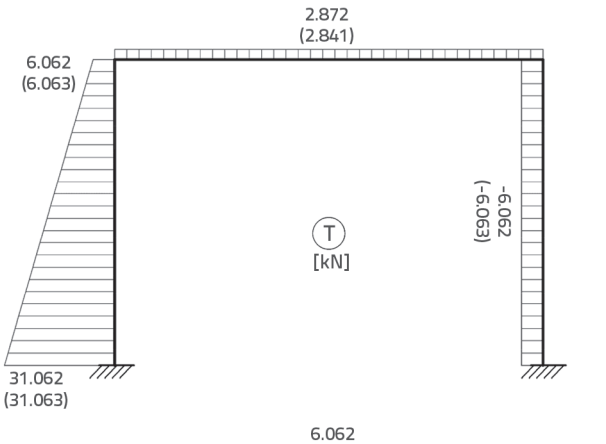


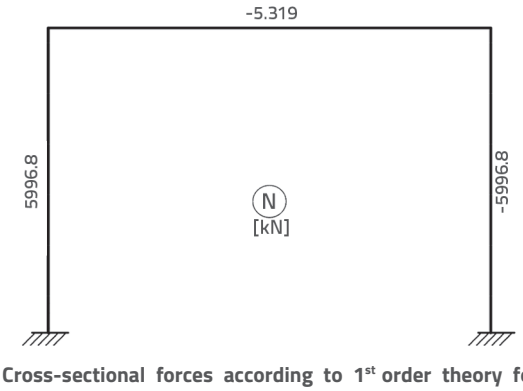
Figure 11. Cross-sectional forces according to 2nd order theory for first frame











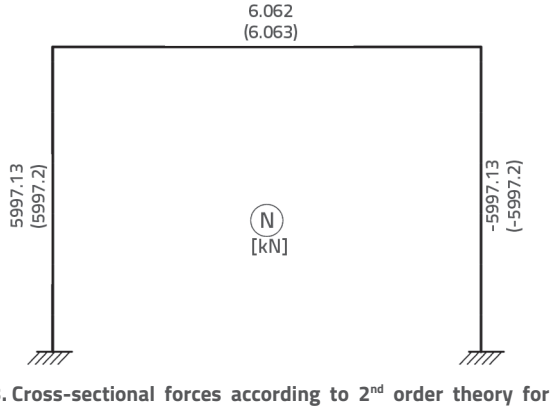


Figure 12. Cross-sectional forces according to 1st order theory for second frame

Figure 13. Cross-sectional forces according to 2nd order theory for second frame

Table 3. Bending moment values for first frame

Beam	Node	Bending moments [kNm]				Difference [%]		
		Theory I order	Theory II order					
		ALIN TOWER	ALIN	TOWER				
				k.e. = štap	k.e. = 1 m			
		/1/	/2/	/3/	/4/	/2/-/1/	/2/-/3/	/2/-/4/
1	1	28,730	**41,437**	37,330	41,429	44,23	**11,00**	0,02
	2	7,175	**17,192**	20,173	17,196	139,61	**-14,78**	-0,02
2	2	-7,175	**-17,192**	-20,173	-17,196	139,61	**-14,78**	-0,02
	3	-11,417	**-22,110**	-23,307	-22,111	93,66	**-5,14**	-0,01
3	3	11,417	**22,110**	23,307	22,111	93,66	**-5,14**	-0,01
	4	15,178	**26,438**	27,683	26,441	74,19	**-4,50**	-0,01

Table 4. Bending moment values for second frame

Beam	Node	Bending moments [kNm]				Difference [%]		
		Theory I order	Theory II order					
		ALIN TOWER	ALIN	TOWER				
				f.e. = štap	f.e. = 1 m			
		/1/	/2/	/3/	/4/	/2/-/1/	/2/-/3/	/2/-/4/
1	1	49,128	**57,111**	61,446	57,120	16,25	**-7,05**	-0,02
	2	-13,223	**-14,449**	-17,207	-14,453	9,27	**-16,03**	-0,03
2	2	13,223	**14,449**	17,207	14,453	9,27	**-16,03**	-0,03
	3	8,981	**5,536**	6,835	5,541	-38,36	**-19,00**	-0,09
3	3	-8,981	**-5,536**	-6,385	-5,541	-38,36	**-13,30**	-0,09
	4	35,576	**25,442**	25,167	25,442	-28,48	**1,09**	0,00

greater differences in results according to the theory of the first and the second order, intensities of concentrated forces amount to about 50 % of the critical load for the first frame.

Values of cross-sectional forces obtained according to theory of the first order using the program ALIN are exactly the same as those obtained by using the program TOWER, Figures 10 and 12. Cross-sectional forces according to the second order theory are presented in Figures 11 (first frame) and 13 (second frame), where the values without brackets correspond to ALIN, while the values in brackets are obtained using TOWER, with division of each beam into 1 m elements.

The following can be seen in Tables 3 and 4: if division of each beam into smaller elements is not performed using TOWER, i.e. if each beam is one finite element, then the differences in bending moments according to the second order theory are up to 15 % for the first frame and up to 19 % for the second frame, compared to "exact" values obtained using program ALIN. In case when division of each beam into 1m finite elements is performed using TOWER, i.e. when vertical beams are divided into 5 segments and horizontal ones into 7 segments, there are practically no differences in bending moment values obtained using ALIN and TOWER (Figures 11 and 13).

5.2. Example 2

A spatial beam clamped at one end, with hinge support at the other end, is subjected to transverse load in xy and xz planes, by concentrated force in the direction of y axis, concentrated couple about z axis, and by axial compression force that is almost one half of the Euler's critical buckling load of the considered beam, see Figure 14.

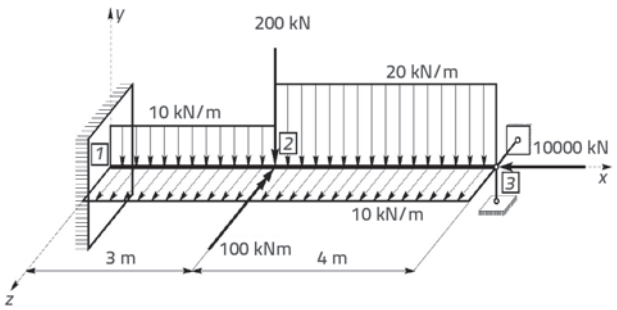


Figure 14. Spatial beam. Geometry and load

It can be seen in Tables 5 and 6 that if no division of each beam into smaller elements is performed using TOWER, i.e. if each beam is one finite element, then the displacement of node 2 according to the second order theory in the direction of y axis is less for 4.22 %, while displacement in the direction of z axis is less for 57.51 %, compared to "exact" values obtained using program ALIN. In this case, bending moment and shear force values according to the second order theory differ for up to 50.53 %, i.e. up to 633.99 %, respectively. In case when all beams are divided into 0.5 m finite elements using Tower, there are practically no differences in values obtained by ALIN and TOWER.

Table 5. Displacement values in direction of axes *y* and *z*

Node	Direction	Displacement [mm]					Difference [%]		
		Theory I order	Theory II order						
		ALIN TOWER	ALIN	TOWER					
				f.e. = štap	f.e.= 1 m	f.e. = 0.5 m			
		/1/	/2/	/3/	/4/	/5/	/2/-/1/	/2/-/3/	/2/-/5/
2	*v*	-16,15	**-31,14**	-29,88	-31,13	-31,14	92,82	**4,22**	0
	w	4,17	**38,29**	24,31	38,20	38,28	818,22	**57,51**	0,03

Table 6 . Bending moment and shear force values

Beam	Node	Internal force	Bending moments [kNm] and hear forces [kN]					Difference [%]		
			Theory I order	Theory II order						
			ALIN TOWER	ALIN	TOWER					
					f.e. = štap	f.e. = 1 m	f.e. = 0.5 m			
			/1/	/2/	/3/	/4/	/5/	/2/-/1/	/2/-/3/	/2/-/5/
1	1	My	61,25	**400,70**	266,20	399,89	400,65	554,20	**50,53**	0,01
		Mz	363,09	**633,92**	614,67	633,83	633,91	74,59	**3,13**	0
		Ty	198,30	**236,99**	234,24	236,98	236,99	19,51	**1,17**	0
		Tz	-43,75	**-92,24**	-73,03	-92,13	-92,24	110,83	**26,30**	0
	2	My	25	**213,91**	151,01	213,52	213,89	755,64	**41,65**	0
		Mz	186,81	**343,45**	341,86	343,44	343,45	83,85	**0,46**	0
		Ty	-168,30	**-206,99**	-204,24	-206,98	-206,99	22,99	**1,35**	0
		Tz	13,75	**62,24**	43,03	62,13	62,24	353,65	**44,64**	0
2	2	My	-25	**-213,91**	-151,01	-213,52	-213,89	755,64	**41,65**	0
		Mz	-286,81	**-443,45**	-441,86	-443,44	-443,45	54,61	**0,36**	0
		Ty	-31,70	**6,99**	4,24	6,98	6,99	122,05	**64,86**	0
		Tz	-13,75	**-62,24**	-43,03	-62,13	-62,24	352,65	**44,64**	0
	3	My	0	**0**	0	0	0	0	**0**	0
		Mz	0	**0**	0	0	0	0	**0**	0
		Ty	111,70	**73,01**	75,76	73,02	73,01	-34,64	**-3,63**	0
		Tz	-26,25	**22,24**	3,03	22,13	22,24	182,98	**633,99**	0

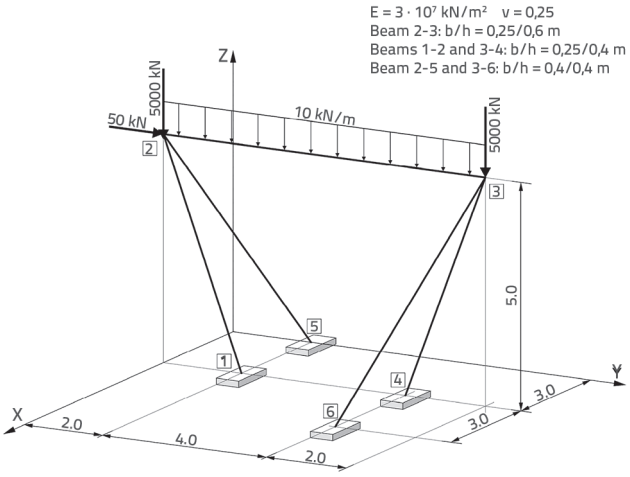


Figure 15. Spatial structure. Geometry and load

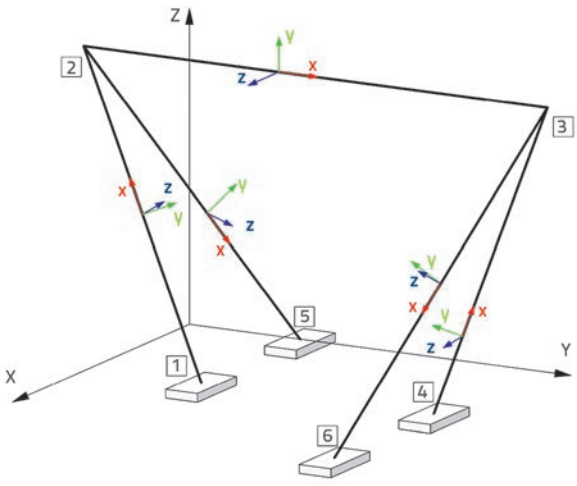


Figure 16. Spatial structure. Local axes of elements

5.3. Example 3

The final example involves analysis of a spatial structure (Figure 15) with non-orthogonal beams, due to given loads and self weight (program ALIN automatically takes into account the self weight, and the same applies to TOWER), as a single load case. Assumed local coordinate axes of various elements of the structure are presented in Figure 16.

Table 7 presents values of bending moments M_z for the considered structure, obtained using programs ALIN and TOWER. When calculation is made according to the second order theory using TOWER, all beams are divided into 0,5 m elements and, in that case, good agreement with ALIN results is obtained.

Figures 17-22 represent plots of cross-sectional forces M_z, M_y and M_t for the considered structure, as obtained using ALIN

Table 7. Values of bending moments M_z for spatial structure

Beam	Node	Bending moments M_z [kNm]			Difference [%]	
		Theory I order	Theory II order			
		ALIN TOWER	ALIN	TOWER f.e.= 0.5 m		
		/1/	/2/	/3/	/2/-/1/	/2/-/3/
1	1	-27,98	-110,86	-110,78	296,21	0,07
	2	-1,82	-71,73	-71,64	3841,21	0,13
2	2	3,95	-74,90	-74,78	-1996,20	0,16
	3	-126,91	-200,25	-200,20	57,79	0,02
3	4	135,14	214,27	214,26	58,55	0,00
	3	106,96	171,03	170,98	59,90	0,03
4	2	-7,79	-0,20	-0,23	-97,43	-13,04
	5	-16,43	-4,89	-4,93	-70,24	-0,81
5	3	-17,78	-22,47	-22,47	26,38	0,00
	6	-34,32	-44,57	-44,62	29,87	-0,11

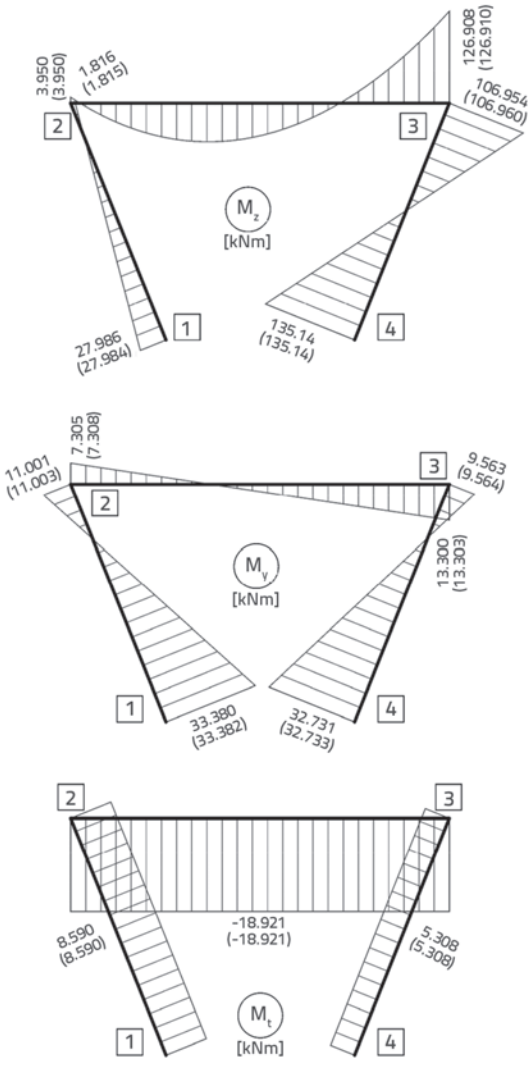


Figure 17. Plots of cross-sectional forces M_z, M_y, M_t according to first order theory for frame 1-2-3-4

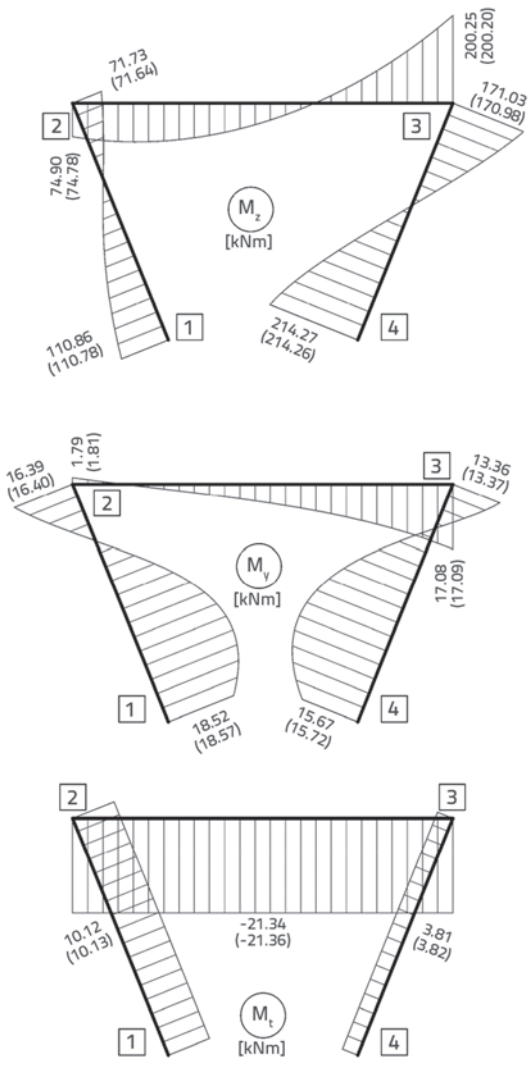


Figure 18. Plots of cross-sectional forces M_z, M_y, M_t according to second order theory for frame 1-2-3-4

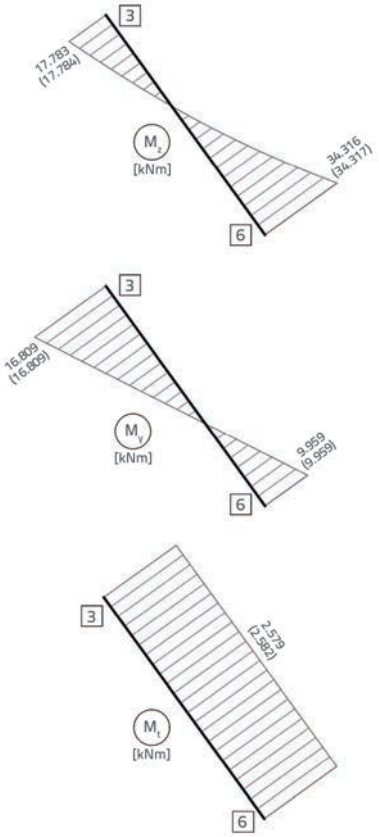


Figure 19. Plots of cross-sectional forces M_z, M_y, M_t according to first order theory for beam 3-6

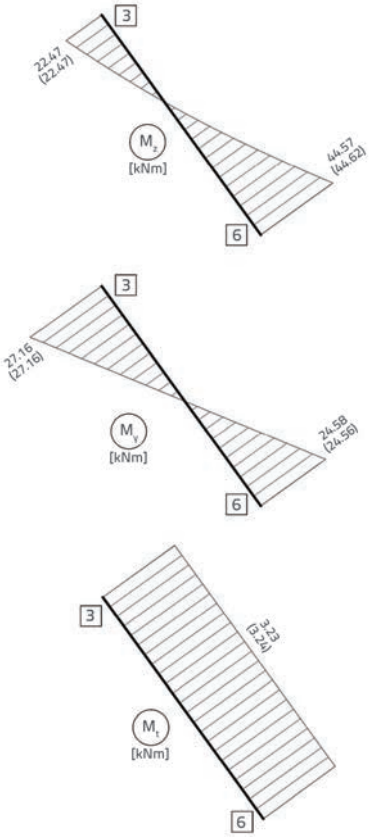


Figure 20. Plots of cross-sectional forces M_z, M_y, M_t according to second order theory for beam 3-6

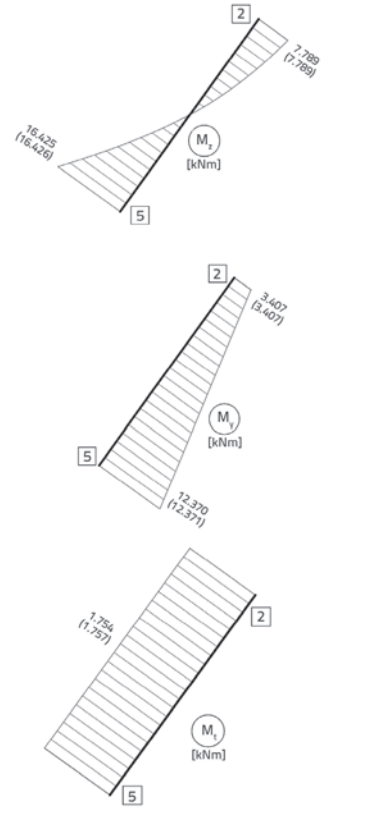


Figure 21. Plots of cross-sectional forces M_z, M_y, M_t according to first order theory for beam 2-5

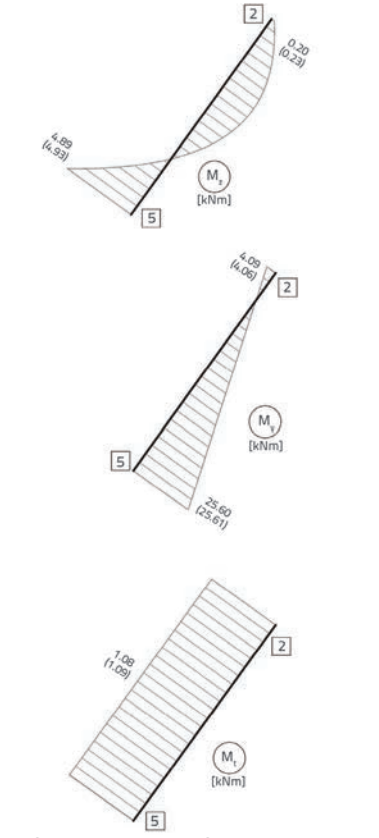


Figure 22. Plots of cross-sectional forces M_z, M_y, M_t according to second order theory for beam 2-5

(values without brackets) and TOWER (values in brackets), according to the theory of the first and second order, for the given load and self weight.

6. Conclusion

The computer code ALIN [5] was developed based on the theory presented in sections 2 and 3. The code is written in programming language C++, and is designed for analysis of spatial and planar framed structures according to the first- and the second-order theories.

Three examples, with comparison of ALIN results with those obtained using the TOWER software, are presented in order to verify the ALIN software.

The results obtained according to the first order theory are in full agreement with results obtained using the TOWER program. On the other hand, differences in results obtained using the ALIN and TOWER programs according to the second order theory are practically inexistent if beams are discretised in TOWER into very small elements. This points to a considerable advantage of ALIN which presents each beam as a single finite element by using the exact stiffness matrix and equivalent load vector.

Therefore, the advantage of ALIN is that beams do not need to be divided into smaller segments when performing analysis according to the second order theory, as is the case when TOWER is used, since ALIN implements the exact stiffness matrix and exact loading vector, as opposed to TOWER, which implements the geometric stiffness matrix.

The ALIN based analysis according to the second order theory is conducted in such a way that, in the first iteration, normal forces are taken as obtained from the previous calculation according to the first order theory, while in the ensuing iterations they are taken as calculated according to the second order theory of previous iteration. The iteration is repeated until the difference between displacements obtained in two successive iterations becomes smaller then some predefined value, or it is stopped after a certain number of iterations. Therefore, as opposed to TOWER [11], where there is only one calculation cycle, the calculation according to ALIN is iterative up to a required level of precision.

Calculation of structures according to the first order theory, where equilibrium conditions are written for a non-deformed structure, may lead to incorrect results when compared to a more accurate theory of the second order, as clearly illustrated through examples given in the paper.

Having in mind that cable-stayed bridges exhibit a geometrically highly nonlinear behaviour, which is the consequence of nonlinear mechanical behaviour of stay cables and the presence of substantial axial forces in the whole structure of the bridge due to tension of stay cables, the second order theory has to be applied in their analysis. The presented theory and this part of ALIN program testing were the basis for numerical analysis of cable-stayed bridges [5]. Cable-stayed bridges are an example of spatial structures which, besides beam elements, also have cable elements, which will be presented in another paper.

REFERENCES

[1] Sekulović, M.: *Teorija linijskih nosača*, Građevinska knjiga, Beograd, 2005.

[2] Sekulović, M.: *Geometrijski nelinearna analiza linijskih nosača, Teorija konstrukcija, Savremeni problemi nelinearne analize*, pp.43-89, Građevinska knjiga, Beograd, 1992.

[3] Bathe, K.J.: *Finite Element Procedures in Engineering Analysis*, Prentice Hall, Englewood Cliffs, New Jersey, 1982.

[4] Čaušević, M.: *Statika i stabilnost konstrukcija - geometrijska nelinearnost*, Školska knjiga, Zagreb, 2003

[5] Žugić, Lj.: *Nelinearna analiza mostova sa kosim kablovima, doktorska disertacija*, Univerzitet u Beogradu, Građevinski fakultet Beograd, 2009.

[6] Brčić, S., Žugić-Zornija, Lj.: Simple and effective matrix-vector library in C++ for non-professionals in computer science, *International Journal of Computational Methods, World Scientific Publishing Company*, 6 (2009) 1, pp. 43-74.

[7] Press W., Teukolsky S., Vetterling W.: *Numerical Recipes in C++: The Art of Scientific Computing*, Cambridge University Press, Third Edition, 2007.

[8] Schildt, H.: *C++ The Complete Reference*, 2nd Ed. Osborne McGraw-Hill, 1995.

[9] Eckel, B.: Misliti na jeziku C++, Mikro knjiga, Beograd, 2003.

[10] http://sourceforge.net/projects/tinyxml/

[11] TOWER 6, *Uputstvo za rad sa programom*, Radimpex, Beograd