CZECH TECHNICAL UNIVERSITY IN PRAGUE
Faculty of Civil Engineering
Department of mechanics
Thákurova 7
166 29 Praha 6

Issue No. 1
Revision No. 0
Date 30.12.2023
Number of pages: 17

Documentation

StableTrussOpt-MATLAB: Program of 3D truss structure optimization with global stability constraints in MATLAB.

MATĚJ LEPŠ
ŠIMON GLANC

Contents

StableTrussOpt-MATLAB is a program of 3D truss structure optimization with global stability constraints in MATLAB. For structural analysis, the frame elements are used to capture true 3D behavior. Then the problem is solved as a sizing optimization task formulated as a SDP program with polynomial constraints.

The software is available at https://github.com/lepsmate/StableTrussOpt-MATLAB. It is provided under LGPL-2.1 license.

Pre-requisites

In addition to Matlab it is essential to have installation of YALMIP (https://yalmip.github.io/) and PENLAB (https://web.mat.bham.ac.uk/kocvara/penlab/) .

Acknowledgment

Financial support for this work was provided by the Technology Agency of the Czech Republic (TAČR), project number FW04020163 (Extension of the HiStruct platform by optimisation of global stability and analysis of design combinations).

Chapter 1

Introduction

StableTrussOpt-MATLAB software is focused on the optimization of truss structures with regard to global stability. Although there are mathematical approaches that enable simultaneous analysis of stability and optimization (SAND), we will separately describe both topics in the introduction part and in the description of the conducted work. First, we will clarify the complexity of stability calculation compared to classical linear analysis. Then, we will describe the nonlinearity of structural optimization including conditions for global stability.

Chapter 2

Linear Stability Analysis

A slender beam subjected to compressive force may fail due to a loss of stability, as shown in Fig.2.1. The simplest case is an ideal beam without imperfections, perfectly straight, loaded centrally. In such a case, linear analysis would not lead to stability loss, and the beam would only bear compression. In reality, there is always some deviation in the shape or location of the load, thus stability assessment is necessary. The stability of a beam can be addressed in several ways. The most well-known approach is the *Geometric Method*, also known as the *Euler Method*. This method calculates the critical force required for the beam to buckle by solving the differential equation of the deflection curve, as depicted in Fig.2.1, and approximating this curve with a sinusoidal function. For the derivation of these influences into the finite element method, we refer the kind reader to the bachelor thesis [7].

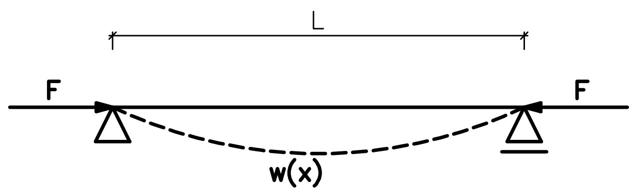

Figure 2.1: The principle of introducing stability theory into a mechanical model.

2.1 Theoretical Introduction

To analyze the stability of structures, standard linear static analysis is no longer sufficient. The influence of internal force distributions on individual members of the structure must be considered using nonlinear analysis, which allows for the inclusion of such factors. However, linear analysis remains an essential component. Its basic element is the stiffness matrix of the structure $\mathbf{K_e}$. It should be noted that the developed program contains a 3D frame element with 12 degrees of freedom. The corresponding stiffness matrix is provided in Appendix A.

The equations for linear analysis of the structure take the form

$$\mathbf{K_e}\boldsymbol{\Delta} = \mathbf{P} \quad , \tag{2.1}$$

where $\boldsymbol{\Delta}$ is the displacement vector and $\mathbf{P}$ is the vector of external forces. Considering the unknown displacements $\boldsymbol{\Delta}$, this equation is *linear*.

However, in transitioning to nonlinear analysis, the fundamental equations are often reduced to a set for which simultaneous equation solving can be utilized. In such cases, the equation takes the form

$$\mathbf{K_t}\delta\boldsymbol{\Delta} = \delta\mathbf{P} \quad . \tag{2.2}$$

Here, $\mathbf{K_t}$ represents the general nonlinear tangent stiffness matrix, which includes both the linearly elastic stiffness matrix and one or more components of nonlinearity, such as incremental deformations or loading. Different levels of nonlinear analysis differ in the definition of the tangent stiffness matrix $\mathbf{K_t}$. In some cases, it is necessary to discretize individual elements of the structure, for example, for linear stability analysis, to better approximate the deformed shape, but the basic form of the equation remains preserved.

Linear stability of the structure introduces the influence of internal forces into the analysis of the structure using the tangent stiffness matrix. The equation is then expressed as

$$[\mathbf{K_e} + \mathbf{K}_\sigma]\delta\boldsymbol{\Delta} = \delta\mathbf{P} \quad , \tag{2.3}$$

where $\mathbf{K}_\sigma$ represents the matrix of initial stresses, also known as the matrix of geometric stiffness. This matrix is used to modify the stiffness matrix depending on the internal forces acting on the structure. It should be noted that Equation 2.3 is essentially *nonlinear* because $\mathbf{K}_\sigma$ is computed for different internal forces than those corresponding to the displacements $\boldsymbol{\Delta}$.

There are several options for assembling this matrix, and for this work, we have chosen the variant provided in Appendix B.

Solving this equation can be readily transformed into a generalized eigenvalue problem, where the equilibrium of the structure in a critical state is expressed as

$$[\mathbf{K_e} + \lambda\mathbf{K}_\sigma^{\mathbf{ref}}]\boldsymbol{\Delta} = 0 \quad , \tag{2.4}$$

where $\mathbf{K}_\sigma^{\mathbf{ref}}$ is the geometric stiffness matrix computed for a specific reference loading, λ (eigenvalue) is the critical load factor corresponding to the given reference loading, and the vector $\boldsymbol{\Delta}$ represents the deformed shape of the structure for each eigenvalue. This equation leads to an eigenvalue problem, the solution of which yields the desired critical load factor.

2.2 Implementation

The problem of stability analysis computation was implemented in the Matlab programming environment. The resulting software is freely available on the GitHub platform:
`https://github.com/lepsmate/StableTrussOpt-MATLAB`.

The source codes are stored in the directory `./framesFEM/`, and their application is demonstrated in the following example. The core of the analysis is the file `criticalLoadFn.m`, which calls the internal Matlab function `eigs` programmed to obtain eigenvalues λ (i.e., critical load multipliers) and their corresponding eigenvectors $\boldsymbol{\Delta}$ (i.e., deflection shapes).

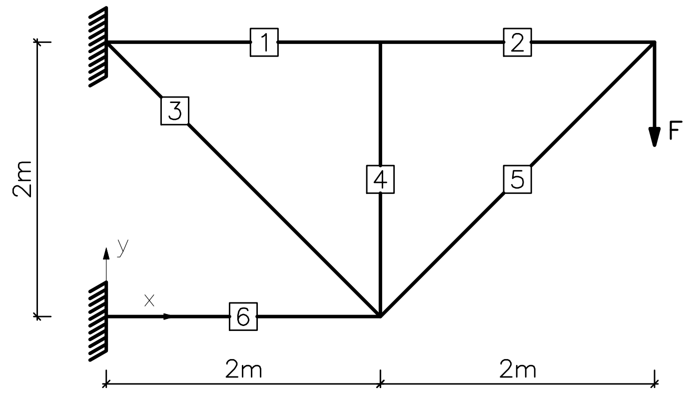

Figure 2.2: Example Setup.

2.3 Example

The studied example is taken from the introduction of the article [5]. The cross-section of the structure in Figure 2.2 is a tubular section with an outer radius of $r_o = 4$ cm and an inner radius of $r_i = 3.5$ cm. The material parameters are the modulus of elasticity $E = 210$ GPa and Poisson's ratio $\nu = 0.3$. Each member is divided into 16 frame elements. Running the file `Torii_ex0_stab1.m` with the implementation of this example yields $P_{crit} = 83.22$ kN. For comparison and validation, results from the OOFEM software package (`http://oofem.org`) are provided. The plot of the corresponding first mode shape is shown in Figure 2.3.

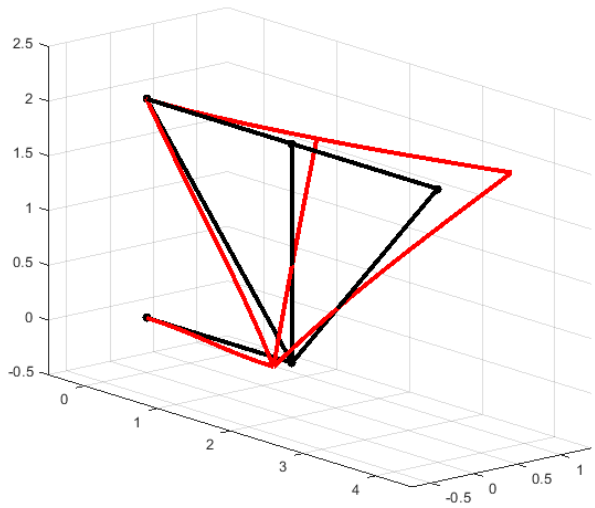

Figure 2.3: First eigenmode shape, showing deflection out of plane.

λ_i	**OOFEM**	**StableTrussOpt-MATLAB**	**[%]**
1	83.21	83.22	0.01
2	288.91	288.95	0.01
3	344.47	344.52	0.01
4	471.79	471.93	0.03
5	521.11	521.18	0.01
6	814.43	814.95	0.06
7	942.06	943.18	0.12
8	1208.39	1209.99	0.13
9	1276.09	1278.20	0.16
10	1729.81	1735.48	0.33

Table 2.1: Comparison of critical load factors for the first 10 eigenmodes.

Chapter 3

Optimization with Respect to Global Stability

If stability analysis alone poses non-trivial challenges, then structural optimization with stability considerations is currently at the forefront of scientific research. Presently, optimizations of truss structures have been solved up to the level of finding global optima. One formulation is the *topological optimization*, where cross-sectional areas can attain zero values, effectively removing the corresponding member from the structure, as formulated in [2]:

$$\min_{\mathbf{A}\in\mathbf{R}^{\mathbf{ne}}} \quad \mathbf{A}^{\mathbf{T}}\mathbf{L} \tag{3.1a}$$

$$\text{s.t.} \quad \left(\mathbf{K_e}(\mathbf{A}) + \bar{\lambda}\mathbf{K_g}\right) \succeq \mathbf{0}, \tag{3.1b}$$

$$A_i \geq 0, \forall i, \tag{3.1c}$$

where $\mathbf{A}$ is the vector of cross-sectional areas, $\mathbf{L}$ is the vector containing the lengths of members, and $\bar{\lambda}$ is the critical factor we aim to achieve. The typical problem is defined with $\bar{\lambda} = 10$, ensuring that the structural stability response is in the linear region. Both matrices $\mathbf{K}$ were introduced in the previous chapter. The mathematical operation $\succeq 0$ represents semi-definiteness of the expression on the left-hand side, which colloquially corresponds to the state where there is no deformed unsupported deflection shape of the structure represented by the matrix $\mathbf{K}_e$ "weakened" by the matrix $\mathbf{K}_g$ (for us, the interesting elements of the matrix $\mathbf{K}_g$ are negative numbers generated by compressive forces).

The entire optimization problem 3.1 is *linear* concerning the sought cross-sectional areas $\mathbf{A}$, but it includes a *nonlinear* constraint 3.1b, which, however, is linear in terms of the areas $\mathbf{A}$. The formulated problem is precisely of the type SAND (*simultaneous analysis and design*), as it simultaneously seeks optimal cross-sectional areas and ensures structural stability. The above problem 3.1 is one of the problems of *semidefinite programming* (SDP), and many mathematical libraries include tools for its solution.

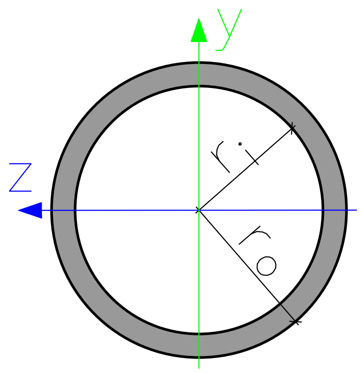

Figure 3.1: Definition of the cross-section for the studied example.

3.1 Optimization of Frame Structures

Let's return to our example in Figure 2.2. For instance, if we consider that individual members are connected by bolted joints and the top member is divided into two parts, then the structure can be analyzed as a truss structure, where each member carries only axial forces, and from the perspective of optimization and design, only the cross-sectional areas matter. However, when the members are welded together and/or the top member is continuous along its entire length, the influence of bending moments on the structure needs to be considered, and thus the effect of resistance to bending and torsion, which is expressed by the moments of inertia of the cross-section. In our example, the cross-section is a tube, as shown in Fig. 3.1. Its area is given by

$$A = \pi {r_o}^2 - \pi {r_i}^2 \tag{3.2}$$

and its moments of inertia are

$$I_y = I_z = \frac{\pi {r_o}^4}{4} - \frac{\pi {r_i}^4}{4}. \tag{3.3}$$

It is apparent that the area and moments of inertia depend on the values of the radii, and that optimization with radii as unknowns would be complex, as discussed further. Here, we make a significant simplification by assuming, for our example, a relationship between the outer and inner radii

$$r_i = 0.875 \cdot r_o \qquad , \tag{3.4}$$

which corresponds to the original stability problem with $r_o = 4$ cm and $r_i = 3.5$ cm. Then, both the area and moments of inertia will depend only on the outer radius, but the radius will still be present to the fourth power in the expression for the moment of inertia. To simplify the optimization problem, the primary unknowns will be chosen as the areas of the cross-sections, and the moments of inertia will be expressed as polynomial approximations depending on the areas. In our specific case, the ratio I/A^2 will be constant, so its approximation is not necessary:

$$\frac{I}{A^2} = \frac{113}{60 * \pi} = const \qquad . \tag{3.5}$$

In another case of, for example, a set of rolled profiles, the approximation of this relationship will be necessary. In our example, the moment of inertia is therefore a quadratic polynomial of areas, $I = \text{const} \times A^2$. For real profiles, third-degree polynomials work well, as shown in Fig. 3.2. Note that we used lower bound approximation for rolled profiles.

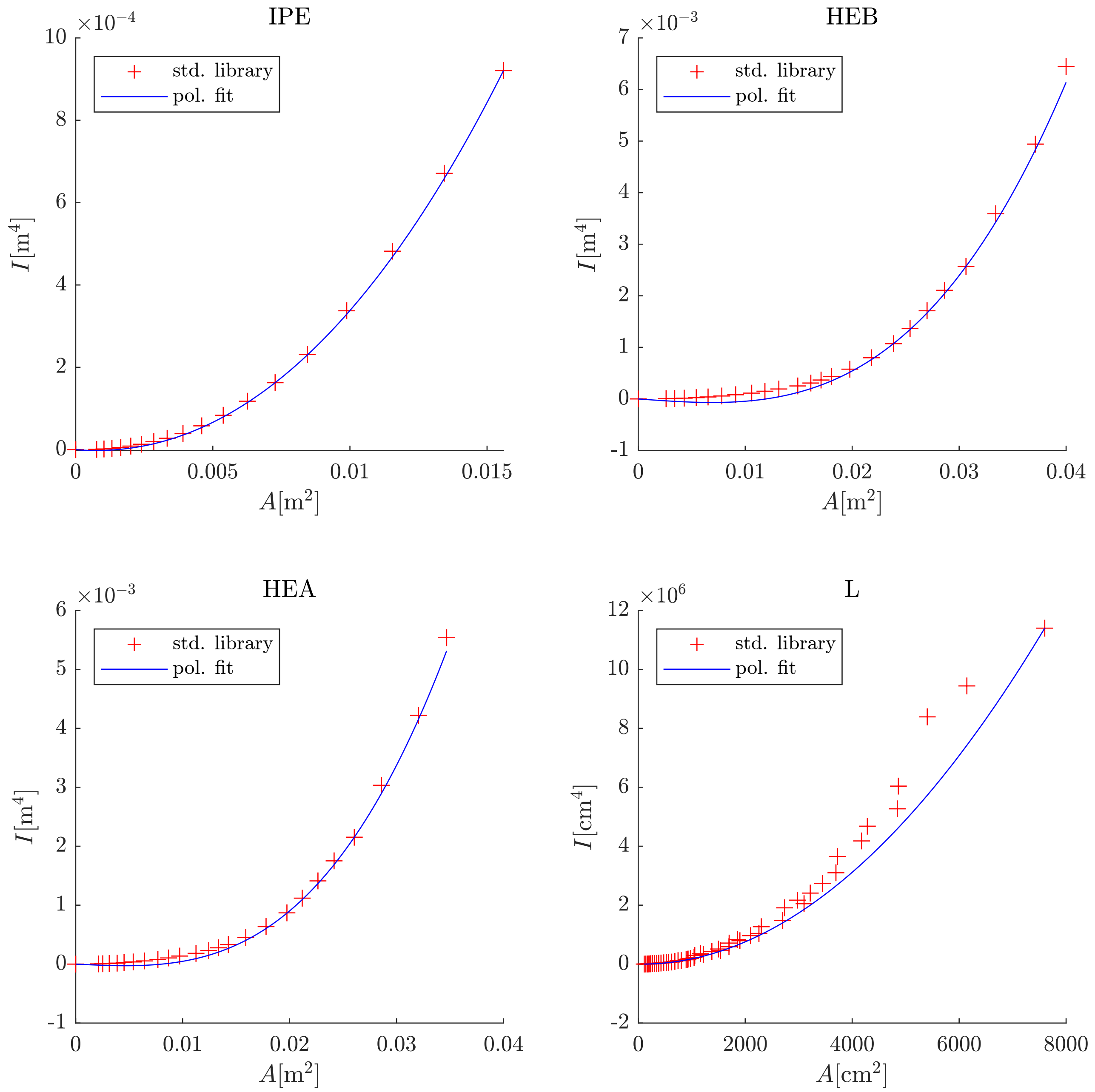

Figure 3.2: Third-degree polynomial approximation of cross-sectional characteristics for four sets of rolled profiles.

So, our new optimization problem will be:

$$\min_{\mathbf{A}\in\mathbf{R}^{\mathbf{ne}}} \quad \mathbf{A}^{\mathbf{T}}\mathbf{L} \tag{3.6a}$$

$$\text{s.t.} \quad \mathbf{I}_{\mathbf{y},\mathbf{z},\mathbf{x}} = \mathcal{P}(\mathbf{A}), \tag{3.6b}$$

$$\left(\mathbf{K}_{\mathbf{e}}(\mathbf{A}, \mathbf{I}_{\mathbf{y},\mathbf{z},\mathbf{x}}) + \bar{\lambda}\mathbf{K}_{\mathbf{g}}\right) \succeq \mathbf{0}, \tag{3.6c}$$

$$A_i \geq A_{min}, \forall i, \tag{3.6d}$$

where Equation 3.6b expresses the polynomial relationship between the areas and moments of inertia. It is worth noting that our project's goal was not to find the optimal topology, but to design an optimally planned structure. Thus, the original "topological" constraint 3.1c has been changed to a constraint of so-called sizing optimization 3.6d, where we do not allow zero cross-sectional areas, but restrict them to the smallest real cross-sectional area A_{min}.

The objective function 3.6a is still *linear*, but the SDP constraint 3.6c is polynomial with respect to the design areas.

In the case of frame structures, only three possible approaches capable of finding global optima were found in the literature. The first approach solves the minimization of mass with displacement constraints by converting the problem into integer programming with quadratic constraints [4]. However, this formulation has significant limitations, limiting the applicability of this method. The second approach in continuous optimization of frame structures involves a series of relaxations using sum of squares (SOS) for specific polynomial semidefinite programming formulations [1], [3]. However, this method does not guarantee finding all global optima in the case of multiple optima. The third possible formulation is introduced by the authors in [6]. This approach uses Lasserre relaxation hierarchy to compute global minimizers, creating lower and upper bounds of the desired solution using polynomial programming. The authors demonstrate that this solution leads to the desired global optimum, but the computational complexity limits this approach to small-scale problems.

Therefore, it can be stated that, according to current knowledge, there is no known formulation suitable for real-sized constructions in the field of sizing optimization of frame structures. Hence, we applied a local approach, which locally linearizes polynomial relationships.

3.2 Implementation

The optimization problem incorporating global stability considerations was implemented in the Matlab programming environment. The YALMIP library (`https://yalmip.github.io/`) was utilized to formulate the optimization problem. This library not only facilitates the definition of variables, objective functions, and constraints but also provides an interface to connect with various optimization solvers. The SDP problem with quadratic constraints was solved using the PENLAB optimization libraries `https://web.mat.bham.ac.uk/kocvara/penlab/`.

3.3 Example

By running the file `Torii_ex0_opt1.m`, the optimization of the example is performed, yielding the targeted $P_{crit} = 10.00$ kN. For comparison and validation, the results from the OOFEM software package (`http://oofem.org`) are provided in the following table.

λ_i	**OOFEM**	**StableTrussOpt-MATLAB**	**[%]**
1	10.00	10.00	0.04
2	37.53	37.50	0.06
3	41.95	41.95	0.02
4	76.01	76.02	0.01
5	84.42	84.34	0.09
6	119.62	119.38	0.20
7	143.41	143.45	0.03
8	229.50	216.41	5.70
9	283.64	229.80	18.98
10	314.50	283.08	9.99

Table 3.1: Comparison of critical load factors

REFERENCES

[1] Masakazu Kojima. Sums of squares relaxations of polynomial semidefinite programs. 12 2003.

[2] Michal Kočvara. On the modelling and solving of the truss design problem with global stability constraints. *Structural and Multidisciplinary Optimization*, 23:189–203, 04 2002.

[3] Kazuo Murota, Yoshihiro Kanno, Masakazu Kojima, and Sadayoshi Kojima. A numerical algorithm for block-diagonal decomposition of matrix -algebras with application to semidefinite programming. *Japan Journal of Industrial and Applied Mathematics*, 27(1):125–160, 2010.

[4] Oguz Toragay, Daniel F. Silva, Alexander Vinel, and Nima Shamsaei. Exact global optimization of frame structures for additive manufacturing. *Structural and Multidisciplinary Optimization*, 65(3):97, 2022.

[5] André Torii, Rafael Lopez, and Leandro Miguel. Modeling of global and local stability in optimization of truss-like structures using frame elements. *Structural and Multidisciplinary Optimization*, 51, 12 2014.

[6] Marek Tyburec, Michal Kočvara, and Martin Kružík. Global weight optimization of frame structures with polynomial programming. *Structural and Multidisciplinary Optimization*, 66(12):257, 2023.

[7] Šimon Glanc. Optimalizace globální stability příhradových konstrukcí s využitím rámových prvků. Bakalářská práce, ČVUT v Praze, Fakulta stavební, Katedra mechaniky, 2022.

Appendix A

Stiffness Matrix

The stiffness matrix of a 3D frame element is taken from [7].

$$\mathbf{K_e} = \mathbf{E}\begin{bmatrix}
\frac{A}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{A}{L} & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{12I_z}{L^3} & 0 & 0 & 0 & \frac{6I_z}{L^2} & 0 & -\frac{12I_z}{L^3} & 0 & 0 & 0 & \frac{6I_z}{L^2} \\
0 & 0 & \frac{12I_y}{L^3} & 0 & -\frac{6I_y}{L^2} & 0 & 0 & 0 & -\frac{12I_y}{L^3} & 0 & -\frac{6I_y}{L^2} & 0 \\
0 & 0 & 0 & \frac{J}{2(1+\nu)L} & 0 & 0 & 0 & 0 & 0 & -\frac{J}{2(1+\nu)L} & 0 & 0 \\
0 & 0 & -\frac{6I_y}{L^2} & 0 & \frac{4I_y}{L} & 0 & 0 & 0 & \frac{6I_y}{L^2} & 0 & \frac{2I_y}{L} & 0 \\
0 & \frac{6I_z}{L^2} & 0 & 0 & 0 & \frac{4I_z}{L} & 0 & -\frac{6I_z}{L^2} & 0 & 0 & 0 & \frac{2I_z}{L} \\
-\frac{A}{L} & 0 & 0 & 0 & 0 & 0 & \frac{A}{L} & 0 & 0 & 0 & 0 & 0 \\
0 & -\frac{12I_z}{L^3} & 0 & 0 & 0 & -\frac{6I_z}{L^2} & 0 & \frac{12I_z}{L^3} & 0 & 0 & 0 & -\frac{6I_z}{L^2} \\
0 & 0 & -\frac{12I_y}{L^3} & 0 & \frac{6I_y}{L^2} & 0 & 0 & 0 & \frac{12I_y}{L^3} & 0 & \frac{6I_y}{L^2} & 0 \\
0 & 0 & 0 & -\frac{J}{2(1+\nu)L} & 0 & 0 & 0 & 0 & 0 & \frac{J}{2(1+\nu)L} & 0 & 0 \\
0 & 0 & -\frac{6I_y}{L^2} & 0 & \frac{2I_y}{L} & 0 & 0 & 0 & \frac{6I_y}{L^2} & 0 & \frac{4I_y}{L} & 0 \\
0 & \frac{6I_z}{L^2} & 0 & 0 & 0 & \frac{2I_z}{L} & 0 & -\frac{6I_z}{L^2} & 0 & 0 & 0 & \frac{4I_z}{L}
\end{bmatrix}$$

Appendix B

Matrix of Initial Stresses

$$\mathbf{K}_\sigma = \frac{\mathbf{F_{x2}}}{\mathbf{L}} \cdot \begin{bmatrix} \mathbf{K}_\sigma^{\mathbf{11}} & \mathbf{K}_\sigma^{\mathbf{12}} \\ \mathbf{K}_\sigma^{\mathbf{21}} & \mathbf{K}_\sigma^{\mathbf{22}} \end{bmatrix}$$

$$\mathbf{K}_\sigma^{\mathbf{11}} = \begin{bmatrix} k_{\text{min}} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{4\kappa_z^2+4\kappa_z+\frac{6}{5}}{\delta_z} & 0 & 0 & 0 & \frac{L}{10\delta_z} \\ 0 & 0 & \frac{4\kappa_y^2+4\kappa_y+\frac{6}{5}}{\delta_y} & 0 & -\frac{L}{10\delta_y} & 0 \\ 0 & 0 & 0 & k_{\text{min}} & 0 & 0 \\ 0 & 0 & -\frac{L}{10\delta_y} & 0 & L^2\frac{\frac{\kappa_y^2}{3}+\frac{\kappa_y}{3}+\frac{2}{15}}{\delta_y} & 0 \\ 0 & \frac{L}{10\delta_z} & 0 & 0 & 0 & L^2\frac{\frac{\kappa_z^2}{3}+\frac{\kappa_z}{3}+\frac{2}{15}}{\delta_z} \end{bmatrix}$$

$$\mathbf{K}_\sigma^{\mathbf{22}} = \begin{bmatrix} k_{\text{min}} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{4\kappa_z^2+4\kappa_z+\frac{6}{5}}{\delta_z} & 0 & 0 & 0 & -\frac{L}{10\delta_z} \\ 0 & 0 & \frac{4\kappa_y^2+4\kappa_y+\frac{6}{5}}{\delta_y} & 0 & \frac{L}{10\delta_z} & 0 \\ 0 & 0 & 0 & k_{\text{min}} & 0 & 0 \\ 0 & 0 & \frac{L}{10\delta_z} & 0 & L^2\frac{\frac{\kappa_y^2}{3}+\frac{\kappa_y}{3}+\frac{2}{15}}{\delta_y} & 0 \\ 0 & -\frac{L}{10\delta_z} & 0 & 0 & 0 & L^2\frac{\frac{\kappa_z^2}{3}+\frac{\kappa_z}{3}+\frac{2}{15}}{\delta_z} \end{bmatrix}$$

$$\mathbf{K}_\sigma^{\mathbf{12}} = \mathbf{K}_\sigma^{\mathbf{21}} = \begin{bmatrix} -k_{\text{min}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{4\kappa_z^2+4\kappa_z+\frac{6}{5}}{\delta_z} & 0 & 0 & -\frac{L}{10\delta_z} \\ 0 & -\frac{L}{10\delta_y} & -\frac{4\kappa_y^2+4\kappa_y+\frac{6}{5}}{\delta_y} & 0 & -\frac{L}{10\delta_y} & 0 \\ 0 & 0 & 0 & -k_{\text{min}} & 0 & 0 \\ 0 & 0 & -\frac{L}{10\delta_y} & 0 & -L^2\frac{\frac{\kappa_y^2}{3}+\frac{\kappa_y}{3}+\frac{2}{30}}{\delta_y} & 0 \\ 0 & -\frac{L}{10\delta_z} & 0 & 0 & 0 & -L^2\frac{\frac{\kappa_z^2}{3}+\frac{\kappa_z}{3}+\frac{2}{30}}{\delta_z} \end{bmatrix}$$

$$k_{\text{min}} = \frac{\min\{K_\sigma(2,2), K_\sigma(3,3), K_\sigma(5,5), K_\sigma(6,6)\}}{1000}$$

B.0.1 Constitutive matrix

$$\mathbf{D} = \frac{\mathbf{E}}{(\mathbf{1}+\mathbf{v})(\mathbf{1}-\mathbf{2v})} \begin{bmatrix} 1-v & v & v & 0 & 0 & 0 \\ v & 1-v & v & 0 & 0 & 0 \\ v & v & 1-v & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2v}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2v}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2v}{2} \end{bmatrix} \tag{B.1}$$

B.0.2 Influence of Shear

$$\kappa_y = \begin{cases} \frac{6 \cdot \mathbf{D}(\mathbf{5},\mathbf{5})}{\mathbf{D}(\mathbf{3},\mathbf{3}) \cdot \mathbf{L}^{\mathbf{2}}}, & \text{pokud } \mathbf{D}(\mathbf{3},\mathbf{3}) \neq \mathbf{0} \\ 0 \end{cases}$$

$$\kappa_z = \begin{cases} \frac{6 \cdot \mathbf{D}(\mathbf{6},\mathbf{6})}{\mathbf{D}(\mathbf{2},\mathbf{2}) \cdot \mathbf{L}^{\mathbf{2}}}, & \text{pokud } \mathbf{D}(\mathbf{2},\mathbf{2}) \neq \mathbf{0} \\ 0 \end{cases}$$

$$\delta_y = (1 + 2 \cdot \kappa_y) \cdot (1 + 2 \cdot \kappa_y)$$
$$\delta_z = (1 + 2 \cdot \kappa_z) \cdot (1 + 2 \cdot \kappa_z)$$