JOURNAL OF CIVIL ENGINEERING AND MANAGEMENT
2010
16(1): 103–121

SAOSYS TOOLBOX AS *MATLAB* IMPLEMENTATION IN THE ELASTIC-PLASTIC ANALYSIS AND OPTIMAL DESIGN OF STEEL FRAME STRUCTURES

Valentinas Jankovski[1], Juozas Atkočiūnas[2]

Department of Structural Mechanics, Vilnius Gediminas Technical University,
Saulėtekio al. 11, 10223 Vilnius, Lithuania
E-mail: [1]JWMsoftcorp@gmail.com; [2]Juozas.Atkociunas@vgtu.lt

Received 19 Mar. 2009; accepted 06 Jan. 2010

Abstract. The improved mathematical model of steel frame structures' design is created. The loading is simple, and plastic strains are evaluated. Energy principles of deformable body mechanics and mathematical programming theory are employed. Equilibrium finite elements with interpolation functions of internal forces are used for discretization. The elements are designed using HE, IPE, RHS steel profile assortments and considering dispersion of geometrical characteristics of profile assortment sets. Optimal design of steel structures is realized by using the experimental tool system *JWM SAOSYS Toolbox v0.42*, which was created by the authors in *MATLAB* environment. *SAOSYS* architecture operates with object-oriented finite elements pseudo-polymorphously. The possibilities of this system are demonstrated by considering a numerical example of optimal design of industrial building frame with strength and stiffness constraints. The assumption of small displacements is adopted for computations.

Keywords: optimal design, energy principle, mathematical programming, steel structures, finite element method, object-oriented programming, *MATLAB*.

1. Introduction

Structural design evaluating plastic strains allows us to exploit carrying capacity of the structure more effectively and create more economical projects (Majid 1974; Atkočiūnas 1999; Choi and Kim 2002; Kaliszky and Logo 2002; Алявдин 2005). It is worth noticing that the assumptions of the limit equilibrium theory are referred to in many papers (Čyras *et al.* 2004; Atkočiūnas *et al.* 2007; Karkauskas 2007). Optimization results under plastic collapse criterion are not decisive in every case because the limit feasibility state of optimal structure can be lost even in the case, when plastic collapse due to excessive inelastic strains and displacements is not achieved (Giambanco *et al.* 1994; Tin–Loi 2000). Therefore, chronologically, the paper (Kaneko and Maier 1981) is most important for the cases, where elastic-plastic structures' optimization with stiffness constraints is analysed.

An optimum criterion of a structure can have energetic or a definite physical meaning, i.e. minimum volume (weight) or minimum cost (Prager 1955; Atkočiūnas *et al.* 2008; Skaržauskas *et al.* 2005; Kalanta *et al.* 2009; Šešok and Belevičius 2008). Engineering development in the field of optimal structural design requires some theoretical and practical knowledge, including the fundamentals of structural mechanics, structural design standards (STR 2.05.08; EN 1993-1-1: 2005) and, finally, modern information technologies. Further, not sticking to chronology, we will mention only some papers (concerning structures' optimization) and provide a comprehensive list of literature. It is mostly the work of M. I. Reitman (1976) in Russian, as well as the books (Brandt 1978; Atrek *et al.* 1984; Lloyd Smith 1990).

It should be noted that the above papers do not take into consideration the relationship between dual theory of mathematical programming and the problems of static and kinematic formulas of rigid deformable body. Namely, the dual theory applied to holonomic plastic deformation process (Koiter 1960) allows the construction of matrices showing the influence of residual internal forces and displacement influence matrices (Atkočiūnas 1994). Finally, the revelation of the mechanical meaning of Kuhn-Tucker optimality constraints (Bazaraa *et al.* 2004) facilitates numerical realization of optimization problems (Atkočiūnas *et al.* 2003–2008). An attempt was made to avoid all these imperfections in the current paper.

An improved mathematical model of minimum volume design of steel structures with plastic strains was created applying energy principles of structural mechanics and mathematical programming theory. Besides, strength, stiffness and stability requirements to structures discretized by finite elements and subjected to local forces were evaluated more accurately. The extreme internal forces of the elements were also restricted by additionally introduced nonlinear yield conditions. There is also a possibility of precise evaluation of extreme element deflections under stiffness conditions (these conditions are mostly the constraints of node displacements of

ISSN 1392–3730 print / ISSN 1822–3605 online
doi:10.3846/jcem.2010.10

the discrete model of the structure). These additional means allow us to avoid densification of the finite elements grid, thereby decreasing the size of the optimization problem, as well as saving computer resources (especially, solving time).

An approximate objective function expressing the volume of a structure is used in the developed mathematical model of the optimization problem. Structural elements are designed considering the dispersion of geometrical characteristics in the sets of profiles and based on the principle of admissible fields of geometrical characteristics of the profiles assortments HE, IPE and RHS (Rectangular Hollow Section).

For practical realization of the optimization problem, the authors created a design algorithm of elastic-plastic structures and structural analysis, as well as optimal design system *JWM SAOSYS Toolbox v0.42* (Structural Analysis and Optimization System) in *MathWorks MATLAB* environment (Jankovski and Atkočiūnas 2008) (Fig. 1). Nonlinear optimization problem considered is nonconvex. The convergence is obtained by an iterative method, i.e. by solving a sequence of nonlinear problems. The system *SAOSYS* combines finite element method, object-oriented programming (OOP) paradigm, mathematical models (Čyras *et al.* 2004) based on structural analysis and extreme energy principles of optimization, as well as mathematical programming theory and methods (Bazaraa *et al.* 2004; Raue *et al.* 2009), principles of the initial structural data input and parameterization, databases of steel profiles' assortments and the output and interpretation methods of textual and graphical data.

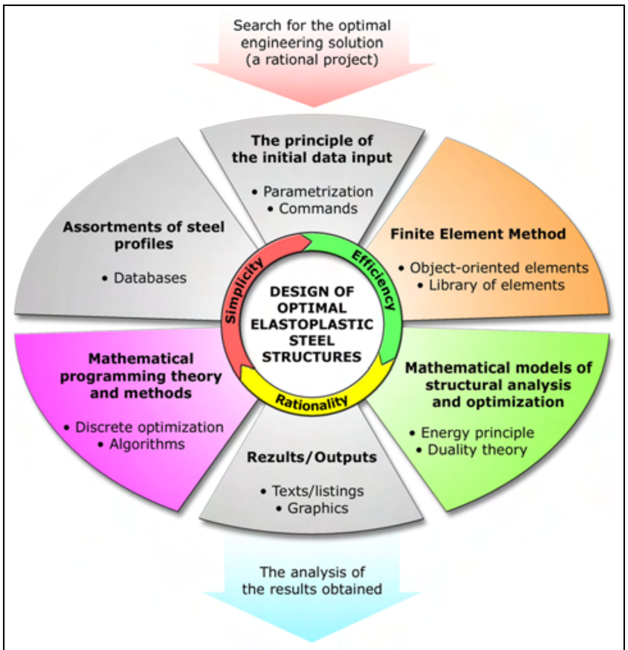

Fig. 1. The main parts of the optimal design system of structures

SAOSYS system's architecture embraces *User mode* pre-post-processor control system, and *Kernel mode*, involving system topology and its components. The created system operates with object-oriented LINK11 and BEAM31 finite elements, composing finite elements' library FELIB, which contains equilibrium and displacement formulations of finite elements for solving various problems. Due to limitations of *MATLAB 7.0* environment OOP facilities, pseudo-polymorphism is realized in *SAOSYS* system which maintains a concept of pseudo-virtual methods. Pre-processor of the created system employs the command data input method of structural model (similar to *ANSYS* software) and a possibility of structural parameterization in a variational design case.

The possibilities of *SAOSYS* system and the proposed technique are illustrated by a numerical example of optimal design of industrial building frame with strength and stiffness constraints. The assumption of small displacements is evaluated for computations.

2. A mathematical model

An elastic-plastic beam structure of known geometry subjected to specified loading is analyzed. Simple loading is perceived as loading, when all loads are proportional to one common factor: thus, plastic deformation holonomicity is indirectly provided. The principle of the minimum energy of A. Haar and von Kármán Th. (Koiter 1960) is valid for this process. The numerical example in this paper is aimed at finding the project of the structure of minimum volume V, whose optimality criterion (1) is provided with respect to strength and stiffness, as well as stability requirements. The optimality criterion consists of the following items: $\mathbf{L}$ is the structural element's length vector; $\mathbf{G}_0$ is the leading vector of the elements' cross-sections of design geometry; $\mathbf{A}(\cdot)$ is the vector function of cross-section geometry conversion to cross-section areas. Thus, minimization is performed when the whole structure's configuration, physical-mechanical characteristics of the material of the elements, loading $\mathbf{F}$ and the vectors of the limiting values of structural nodes' displacements $\mathbf{u}_{min}$, $\mathbf{u}_{max}$ and ultimate deflections $\mathbf{v}_{min}$, $\mathbf{v}_{max}$ of the elements are given. The vectors $\mathbf{v}_{min}$, $\mathbf{v}_{max}$ are used to control stiffness conditions.

The basis of this optimization is a system of equations and dependencies, defining a real stress-strain state of an elastic-plastic structure before plastic failure. This system can be created assuming the constraints of the static formulae of the analysed problem and Kuhn-Tucker's optimality conditions of the problem (Atkočiūnas and Merkevičiūtė 2003). The above-mentioned system of dependencies (giving the results of influence matrices $[Z]$ and $[Y]$ of residual internal forces $\mathbf{S}_r$ and residual displacements $\mathbf{u}_r$ respectively) is often referred to as a generalized Lagrange problem. Further, the internal forces and displacements at the elastic stage will be denoted as $\mathbf{S}_e$ and $\mathbf{u}_e$, respectively (16, 15), where $[\beta]$ is the matrix of the influence of structural displacements. Thus, the aim of the computations is to find optimal distribution of geometrical characteristics of the elements' cross-sections $\mathbf{G}_0$, while safe exploitation of the structure with plastic strains is secured.

Thus, a mathematical model for the problem of the minimal volume is as follows:

find $\min V = \mathbf{L}^T \cdot \mathbf{A}(\mathbf{G}_0)$ (1)

subject to: $[B]\mathbf{G}_0 - [Z]\boldsymbol{\lambda} \geq \mathbf{b}$, (2)

$$\boldsymbol{\lambda}^T\left([B]\mathbf{G}_0 - [Z]\boldsymbol{\lambda} - \mathbf{b}\right) = 0, \quad (3)$$

$$[Y]\boldsymbol{\lambda} \geq \mathbf{u}_{min} - \mathbf{u}_e, \quad (4)$$

$$[Y]\boldsymbol{\lambda} \leq \mathbf{u}_{max} - \mathbf{u}_e, \quad (5)$$

$$\boldsymbol{\Psi}_{S,ext}(\boldsymbol{\lambda}, \mathbf{S}_e) \leq \mathbf{0}, \quad (6)$$

$$\mathbf{v}_{min} \leq \boldsymbol{\Psi}_{v,ext}(\boldsymbol{\lambda}, \mathbf{u}_e) \leq \mathbf{v}_{max}, \quad (7)$$

$$\boldsymbol{\lambda} \geq \mathbf{0}, \quad (8)$$

$$\mathbf{G}_0 \geq \mathbf{G}_{0,min}. \quad (9)$$

Thus, the mathematical model (1)–(9) for the volume minimization problem of elastic plastic beam structure with variables $\mathbf{G}_0$ and $\boldsymbol{\lambda}$ consists of nonlinear objective function (1) and constraints-conditions, such as linear inequalities (2, 8); nonlinear complementary slackness condition (3); stiffness constraints (4, 5); additional nonlinear yield conditions (6), evaluating extreme internal forces of elements; nonlinear stiffness constraints (7), evaluating extreme deflections of elements; and constructional constraints (9). Furthermore, the following notation is used in the given mathematical model:

$$[\beta] = \left([A][K][A]^T\right)^{-1}, \quad (10)$$

$$[H] = [\alpha][A][K], \quad (11)$$

$$[Q] = \left([K][A]^T[H] - [K]\right)[\Phi]^T, \quad (12)$$

$$[Z] = [\Phi][Q], \quad (13)$$

$$[Y] = [H][\Phi]^T, \quad (14)$$

$$\mathbf{u}_e = [\beta]\mathbf{F}, \quad (15)$$

$$\mathbf{S}_e = [K][A]^T\mathbf{u}_e, \quad (16)$$

$$\mathbf{b} = [\Phi]\mathbf{S}_e. \quad (17)$$

Then, the vectors of the total displacements $\mathbf{u}$ and internal forces $\mathbf{S}$ of the structure are as follows:

$$\mathbf{u} = \mathbf{u}_e + \mathbf{u}_r = \mathbf{u}_e + [Y]\boldsymbol{\lambda}, \quad (18)$$

$$\mathbf{S} = \mathbf{S}_e + \mathbf{S}_r = \mathbf{S}_e + [Q]\boldsymbol{\lambda}. \quad (19)$$

It only remains to note that structural configuration matrix $[B]$ and the vector of plastic multipliers $\boldsymbol{\lambda}$ are included into the direct yield conditions (2) (the vector $\mathbf{b}$ is known during the iterative process). The vectors $\mathbf{u}_{min}$, $\mathbf{u}_{max}$ are interpreted as maximum vectors of negative and positive values of the restricted nodes' displacements; vectors $\mathbf{v}_{min}$, $\mathbf{v}_{max}$ are maximum vectors of negative and positive values of elements' deflections; $[A]$ is the matrix of coefficients of equilibrium equations of the structure; $[K] = [D]^{-1}$ is the stiffness matrix of structural elements; $[\Phi]$ is the matrix of structural elements' linear yield conditions at the nodes.

3. Beam finite elements: yield and strength conditions

Steel structures will be modelled by the equilibrium finite elements with interpolation functions of internal forces (De Veubeke 1963; Gallagher 1975; Kalanta 1995; Wilson 2002). LINK11 and BEAM31 types of finite elements are described in this paper. All elements $k = 1, 2, 3, ..., n_e$ of the structure compose a set K of finite elements. Subsets K_{11} and K_{31}, corresponding to finite element types LINK11 and BEAM31, compose the set K. The subset R_r is composed of the elements of the same type, material and cross-section's geometrical characteristics. The set R is composed of the subsets R_r of attributes. After describing the sets of the finite elements of structural model, we will discuss yield and strength conditions of every element in detail.

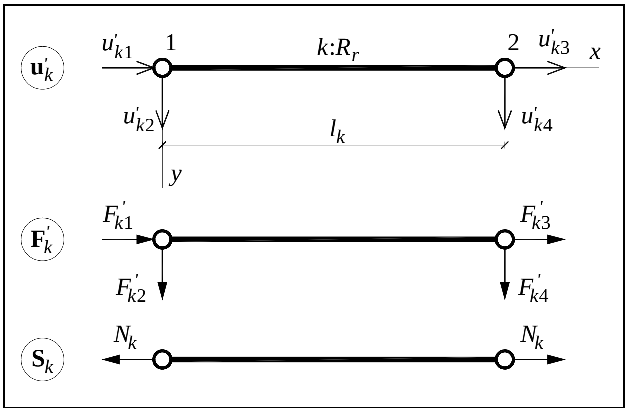

Fig. 2. Finite element LINK11

LINK11. It is an elastic-plastic equilibrium finite element of truss which can only lengthen or shorten (i.e. strains in axial direction are only evaluated) (Fig. 2). The vector of the internal forces of the element is $\mathbf{S}_k = \{N_k\}$ and its yield-strength conditions (i.e. strength and stability requirements) are as follows:

$$\begin{cases} {}^{11}\phi_1 = N_k - f_{y,k}A_{0,k} \leq 0, \\ {}^{11}\phi_2 = -N_k - \varphi_k f_{y,k}A_{0,k} \leq 0, \end{cases} \quad k \in K_{11}. \quad (20)$$

The first constraint is the yield condition of the tensioned element section, since the second is the strength-stability condition of the compressed element (STR 2.05.08. 2005). Buckling reduction factor φ_k of centrally compressed beam is calculated as follows:

$$\varphi_k = \varphi\left(E_k, f_{y,k}, l_k/i_k\right), \quad (21)$$

$$i_k = \sqrt{\frac{I_k}{A_{0,k}}}, \quad I_k = \min\left\{I_{y,k}, I_{z,k}\right\}. \quad (22)$$

The yield-strength conditions (20), written in matrix-vector form are as follows:

$$[\Phi_k]\mathbf{S}_k - A_{0,k}\mathbf{B}_k\left(A_{0,k}, I_k\right) \leq \mathbf{0}, \quad k \in K_{11}. \quad (23)$$

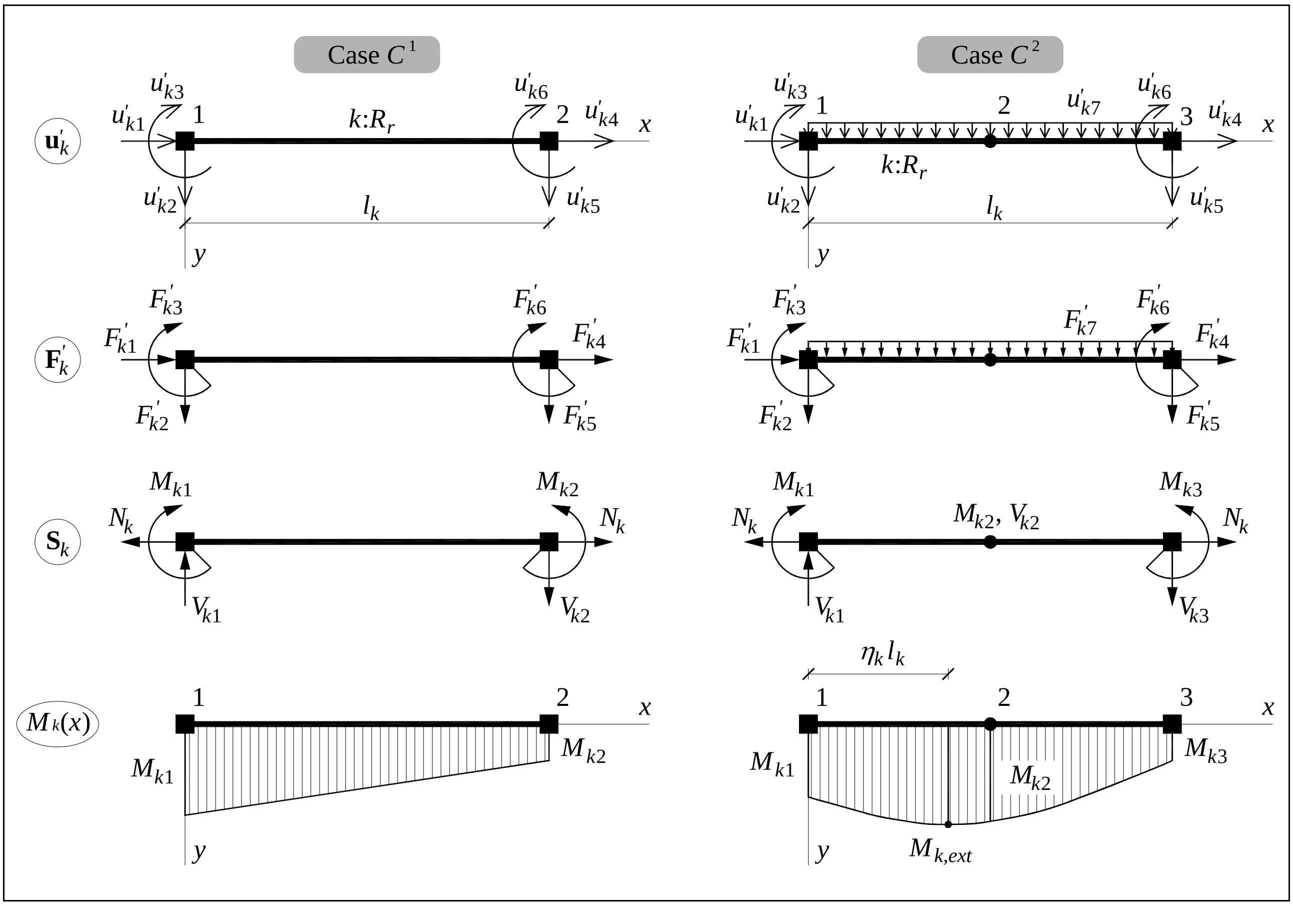

Fig. 3. Finite element BEAM31

Here, N_k is the element's axial force; $f_{y,k}$ is tensile steel strength depending on the yield stress; E_k is elasticity modulus of steel; l_k is the element's length; $A_{0,k}$, I_k, i_k are the element's cross-section area (design parameter), the inertia moment and inertia radius of cross-section, respectively; $[\Phi_k]$, $\mathbf{B}_k(\cdot)$ is the matrix of yield-strength conditions ratios and configuration vectorial function.

BEAM31. It is an equilibrium finite element of the 2D beam under bending and tension or compression (Fig. 3). Two cases of this element are considered: C^1 is an element subjected only to nodal loads (default); C^2 is an element subjected to nodal loads and uniformly distributed load F'_{k7}. Then, the vector of the element's internal forces $\mathbf{S}_k$ is as follows:

$$\mathbf{S}_k = \begin{vmatrix} \{M_{k1}, M_{k2}, N_k\}, & C^1; \\ \{M_{k1}, M_{k2}, M_{k3}, N_k\}, & C^2. \end{vmatrix} \qquad (24)$$

Yield conditions must be satisfied in the nodes of the BEAM31 finite element:

$$\begin{cases} {}^{31}\phi_1(M_{ki}, N_k) = \xi|M_{ki}| + c_k|N_k| - f_{y,k}W_{pl,y,0,k} \le 0, \\ {}^{31}\phi_2 = |M_{ki}| - f_{y,k}W_{pl,y,0,k} \le 0, \end{cases} \qquad (25)$$

$$c_k = W_{pl,y,0,k} / A_k, \qquad (26)$$

$$i = 1, 2, \langle 3 \rangle_{C^2}; \quad k \in K_{31}.$$

These conditions written in the matrix-vector form are as follows

$$\left[\Phi_k\left(W_{pl,y,0,k}, A_k\right)\right]\mathbf{S}_k - W_{pl,y,0,k}\mathbf{B}_k \le \mathbf{0}, \qquad (27)$$

where M_{ki} denotes bending moments in the element's nodes; $\xi = 0{,}85$ is the ratio; $W_{pl,y,0,k}$, A_k is plastic section modulus of the element's cross-section (design parameter) and cross-section area, respectively.

The extreme bending moment $M_{k,ext}$ should be evaluated while designing elements of C^2 case (i.e. when bending moments vary according to the second degree curve) (Fig. 3). It can be implemented approximately by increasing the number of finite elements. The second way of accurate calculation, which will be discussed below, is the direct application of additional nonlinear yield conditions (6). With reference to the formula (19), the true vector of internal forces $\mathbf{S} \equiv \{\mathbf{S}_k, k \in K\}$ can be calculated. A relative position η_k of the feasible extreme bending moment is expressed as follows:

$$\eta_k = \begin{vmatrix} \frac{1}{4}\frac{d_{1k}}{d_{2k}}, & d_{2k} \ne 0; \\ \varnothing, & \text{otherwise}, \end{vmatrix} \qquad (28)$$

where

$$d_{1k} = 3M_{k1} - 4M_{k2} + M_{k3}, \quad (29)$$
$$d_{2k} = M_{k1} - 2M_{k2} + M_{k3}. \quad (30)$$

If $\eta_k \in (0; 1)$, the value of the extreme bending moment is calculated as follows

$$M_{k,ext} = M_{k1} - \frac{1}{8}\frac{d_{1k}^2}{d_{2k}}. \quad (31)$$

Finally, the additional nonlinear yield condition (25) of the element with reference to the extreme bending moment $M_{k,ext}$ is expressed as

$$\Psi_{S,ext,k} = {}^{31}\phi_1\left(M_{k,ext}, N_k\right) \le 0, \quad k \in K_{31}. \quad (32)$$

4. Stiffness condition of the BEAM31 finite element

The stiffness conditions (4, 5) in the structural volume minimization problem allow us only to restrict the real (total) displacements **u** of the nodes of the discrete structural model. In order to check extreme deflections $\mathbf{v}_{k,ext}$ (Fig. 4) of separate elements, we must densify the grid of finite elements, or calculate accurately, i.e. directly apply stiffness conditions (7), corresponding to extreme deflections. We will discuss it in more detail. Applying the conditions (18) we begin to calculate the vector **u** of global displacements, from which we pick the values of nodes' displacements $\mathbf{u}_k$ of a single element.

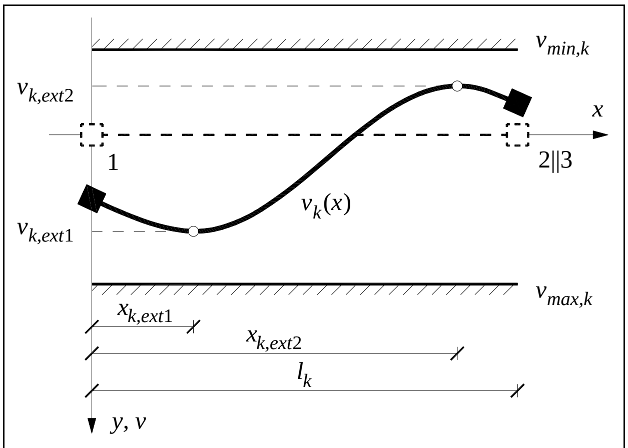

Fig. 4. Extreme deflections of the BEAM31 element

To create the interpolation function of the element's deflections $v_k(x)$, we choose a third-degree polynomial:

$$v_k(x) = \left[N_{v,k}(x)\right]\mathbf{u}'_k = \left[N_{v,k}(x)\right]\left[T_k\right]\mathbf{u}_k, \quad (33)$$

where $[N_{v,k}(x)]$ is function-matrix of the element's shape; $[T_k]$ is transformation matrix of the displacements of the element's nodes; $\mathbf{u}_k = \{u_{k1}, u_{k2}, ..., u_{k6}, <u_{k7}>_{C^2}\}$ is global displacement vector of the element's nodes; $\mathbf{u}'_k$ is local displacement vector of the element's nodes. For deflection interpolation function $v_k(x)$ we apply a stationary condition. Then, while solving the quadratic equation, we try to find two solutions to the locations of extreme deflections $x_{k,ext,i}$:

$$\mathbf{x}_{k,ext} \equiv \begin{vmatrix} x_{k,ext1} \\ x_{k,ext2} \end{vmatrix} = \begin{vmatrix} \dfrac{-b \pm \sqrt{D}}{2a}, & D \ge 0; \\ \varnothing, & D < 0, \end{vmatrix} \quad (34)$$

where

$$D = b^2 - 4au'_{k3}, \quad (35)$$

$$a = \frac{3}{l_k^3}\left(2u'_{k2} + l_k u'_{k3} - 2u'_{k5} + l_k u'_{k6}\right), \quad (36)$$

$$b = \frac{2}{l_k^2}\left(-3u'_{k2} - 2l_k u'_{k3} + 3u'_{k5} - l_k u'_{k6}\right). \quad (37)$$

The extreme deflections of the element are calculated for each location $x_{k,ext,i}$:

$$\mathbf{v}_{k,ext} : v_{k,ext,i} = \begin{vmatrix} v_k\left(x_{k,ext,i}\right), \; x_{k,ext,i} \in \left(0; l_k\right); \\ \varnothing, \quad \text{otherwise,} \end{vmatrix} \quad i = 1, 2. \quad (38)$$

Finally, these nonlinear stiffness conditions (which evaluate extreme deflections and are denoted in the mathematical model (1)–(9) as (7)) are expressed as follows:

$$v_{min,k} \le \left[\Psi_{v,ext,k} = \begin{smallmatrix}\min\\ \max\end{smallmatrix} \mathbf{v}_{k,ext}\right] \le v_{max,k}, \quad k \in K_{31}. \quad (39)$$

5. Assortments: fields of discrete geometrical charac-teristics of the profiles in structural optimization

Steel structures' design is closely connected to discrete sets of profiles' assortments. Analysing the distribution of geometrical characteristics of the profiles IPE, HE, RHS (Fig. 5, Fig. 6), we can see that the single-valued dependence between cross-section geometrical characteristics A-$W_{pl,y}$ and I-A does not exist. Therefore, the admissible points ${}^k\mathbf{G}$ (Fig. 6) are to be found in discrete fields $\mathscr{D}_{A-W_{pl,y}}$ and $\mathscr{D}_{I-A}$ of assortments during the optimization process.

In the case of the elements of different types in a structure, the geometry vector ${}^k\mathbf{G}$ of cross-section takes the form of:

$${}^k\mathbf{G} \equiv \left\{G_{0,k}, G_{1,k}\right\} \equiv \begin{vmatrix} \left\{A_{0,k}, I_k\right\}, & k \in K_{11}; \\ \left\{W_{pl,y,0,k}, A_k\right\}, & k \in K_{31}, \end{vmatrix} \quad k \in K. \quad (40)$$

For the whole structure, the following notation is correct:

$$[G] \equiv \left[\mathbf{G}_0, \mathbf{G}_1\right]. \quad (41)$$

The mathematical model (1)–(9) involves only $\mathbf{G}_0$ design parameter (the problem's variable), which is the vector of the leading design geometry. The inertia moments I_k and areas A_k of cross-sections compose the vector of the driven geometry $\mathbf{G}_1$. While solving the optimization problem (1)–(9) by the iteration process, the leading geometry is optimized, whereas the lagged driven geometry is only corrected with reference to the yield conditions (2, 6) and admissible field bounds $A_{min}(W_{pl,y}) - A_{max}(W_{pl,y})$, $I_{min}(A) - I_{max}(A)$ of the assortments.

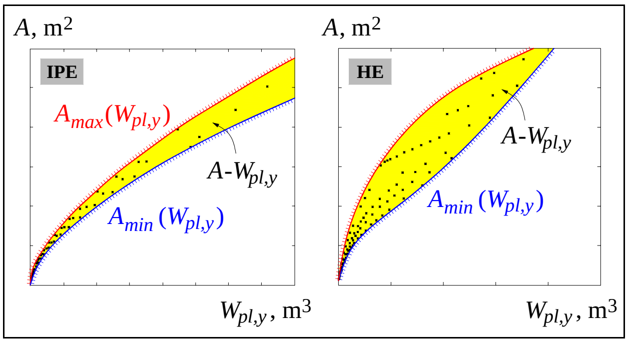

Fig. 5. Admissible fields of the discrete characteristics of IPE and HE profiles A-$W_{pl,y}$

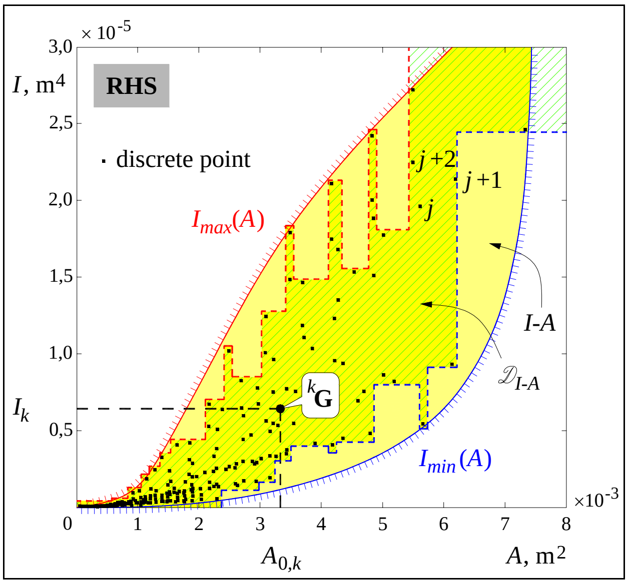

Fig. 6. Fields of discrete values $\mathscr{D}_{I\text{-}A}$ and admissible I-A characteristics of RHS profile assortment

6. Optimality criterion of the structure

The nonlinear objective function (1) reflects the volume V of the structural elements written using the vector of the leading cross-section geometry $\mathbf{G}_0$:

$$\mathbf{A}(\mathbf{G}_0): A_k(G_{0,k}) = \begin{cases} G_{0,k}, & k \in K_{11}; \\ \bar{A}_k(G_{0,k}), & k \in K_{31}, \end{cases} \quad k \in K. \qquad (42)$$

Functions $A_k(\cdot)$ relate to different geometrical cross-section characteristics of the elements ($G_{0,k}$: $A_{0,k}$ | $W_{pl,y,0,k}$) with the areas A_k of these cross-sections. Because of the lack of a single-valued dependency between $A(W_{pl,y})$ (the case of the BEAM31 element design) (Fig. 5), we analyzed two appropriate solving methods of this problem: 1. approximate mean curves of profile assortments; 2. isocurve fields, approximating admissible fields of profile assortments A–$W_{pl,y}$.

The numerical experiments showed that the first method proved to be better because a more optimal solution was reached. The second method 'disbalances' the optimization problem during the iteration process and, therefore, the solution departs from the optimal version.

Approximative mean curves. For approximating discrete points' distribution A–$W_{pl,y}$ of the profile assortments HE and IPE we choose a third-degree polynomial

$$\bar{A}_k(W_{pl,y}) = a_1 W_{pl,y}^3 + a_2 W_{pl,y}^2 + a_3 W_{pl,y} + a_4, \quad (43)$$

the ratios a_i of which are derived by creating the following non-correlation function of the least squares method

$$s(A, W_{pl,y}) = \left[A - \bar{A}_k(W_{pl,y})\right]^2. \quad (44)$$

We apply stationary conditions for the total non-correlation of a set of discrete points $\mathscr{D}$ and a condition for the polynomial, approximating the edge point dependency of a discrete set. Then, four-equation system takes the form of:

$$\begin{cases} \dfrac{\partial}{\partial a_i} \sum\limits_{j \in \mathscr{D}} s(A_j, W_{pl,y,j}) = 0, & i = 1, 2, 3; \\ s({}^{\mathscr{D}}A_{min}, {}^{\mathscr{D}}W_{pl,y,min}) = 0. \end{cases} \quad (45)$$

We solve the given equation system for separate profile assortments at the point of the ratios a_i (i = 1, 2, ..., 4). The obtained ratios' values of approximative polynomials (Fig. 7) are presented in Table 1.

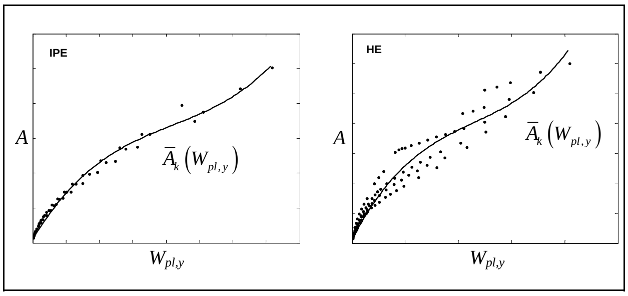

Fig. 7. Approximative mean curves of IPE and HE profiles

Table 1. Ratios of approximative polynomials

i	a_i		
	HE	IPE	HE ∪ IPE
1	$1{,}3331 \cdot 10^4$	$1{,}1407 \cdot 10^5$	$1{,}2391 \cdot 10^4$
2	$-4{,}5793 \cdot 10^2$	$-1{,}4351 \cdot 10^3$	$-4{,}2690 \cdot 10^2$
3	6,9108	7,9208	6,7065
4	$1{,}1582 \cdot 10^{-3}$	$4{,}8818 \cdot 10^{-4}$	$5{,}1086 \cdot 10^{-4}$

Fields of approximative isocurves. To avoid the problems related to the above-mentioned not existing single-valued dependencies of $\bar{A}_k(W_{pl,y})$, we analyzed the fields of the isocurves, approximating the admissible profile fields $\mathscr{D}$ (Fig. 8).

The function of isocurves' field is linearly interpolated between two *spline* functions $A_{min}(\cdot)$ and $A_{max}(\cdot)$, which bind the admissible field $\mathscr{D}$. This function may be written as follows

$$\bar{A}_k(W_{pl}, \eta) = \left[A_{max}(W_{pl}) - A_{min}(W_{pl})\right]\eta + A_{min}(W_{pl}). \quad (46)$$

We solve the optimization problem (1)–(9) by iterations and control the dependence of the ${}^{k}\mathbf{G} \equiv \{W_{pl,y,0,k}, A_k\}$ points on the admissible field. During this process, we correct relative ratios of the approximative isocurves (46) selectors:

$$\eta_k = \frac{A_k - A_{min}\left(W_{pl,y,0,k}\right)}{A_{max}\left(W_{pl,y,0,k}\right) - A_{min}\left(W_{pl,y,0,k}\right)}, \quad k \in K_{31}. \quad (47)$$

Then, the optimization problem developed for a new iteration is solved with the new corrected approximative functions $\bar{A}_k\left(G_{0,k}\right) = \bar{A}_k\left(G_{0,k}, \eta_k\right)$, which compose the objective function (1).

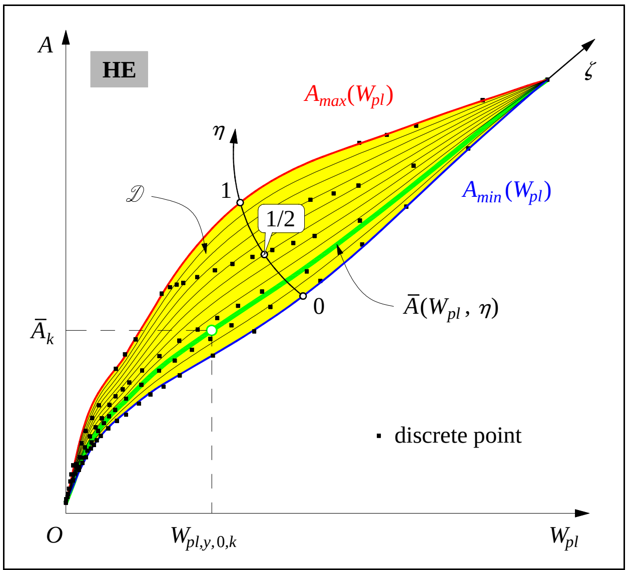

Fig. 8. Isocurves' field of HE profile assortment

7. Design algorithm

Design algorithm of elastic-plastic structures is realized in the integrated design system, which combines *MATLAB* and system *SAOSYS* developed by the authors. Further, we will describe the main parts of the algorithm (Fig. 9).

Structural modelling. Parameterization (in the case of variant design) and discretization of a structure with elastic-plastic equilibrium beam LINK11 and BEAM31 finite elements are performed. The initial data file of the structure (*SAOSYS* pre-processor Batch and Data file – BDF) is created.

Preparing design environment. The routine P1 of *SAOSYS* pre-processor reads and translates BDF, as well as creating the database DB of structural finite element model. The routine P2, which controls profile assortments, reads and prepares the assortment of steel profiles HE, IPE, RHS from SRT database. The routine P3 loads the finite elements' library FELIB of the *SAOSYS* system. Finally, the routine P4 creates and initializes the assemblage FE of the finite elements, which compose the structure (i.e. calculates the length of the elements l_k and direction cosine vectors $\mathbf{n}_k$, as well as preparing displacement compatibility vectors and performing other operations).

Preliminary calculations. The routine P5 directly creates the ratios' matrix $[A]$ of the structure from separate finite element matrices $[A_k]$ of assemblage FE and external loading matrix $[F]$ (in the case of single loading – vector $\mathbf{F}$). The boundary conditions of a discrete model are evaluated. The routine P6 activates the yield conditions of the elements of FE assemblage. The edge values' vectors $\mathbf{G}_{0,min}$, $\mathbf{G}_{0,max}$, $\mathbf{G}_{1,min}$, $\mathbf{G}_{1,max}$ of the leading and driven geometry with reference to SRT database of profile assortments are created. The total length vector $\mathbf{L}$ of the sets of structural elements and element length vector $\mathbf{L}_{max}$ composed of the longest elements in groups (R_r, $r \in R$) are created. The vectors of the nodes' limit displacements $\mathbf{u}_{min}$, $\mathbf{u}_{max}$ and the element's ultimate deflections $\mathbf{v}_{min}$, $\mathbf{v}_{max}$ are also created.

Solving the optimization problem. To solve the optimization problem (1)–(9), we use an iterative approximation and begin with the highest geometrical values of the vectors $\mathbf{G}_0 = \mathbf{G}_{0,max}$, $\mathbf{G}_1 = \mathbf{G}_{1,max}$.

- *Step 1*: the routine P7, with reference to the created geometrical matrix $[G]$ (41) of $W_{pl,y,0,k}$ and A_k values, performs an interpolation operation $I_{y,k} = I_y(\cdot)$ of the cross-sections' inertia moments of the BEAM31 finite elements. This operation is described in more detail in Section 7.1. The created vector $\mathbf{I}_y$ of interpolated inertia moments is used for constant flexibility matrices $[D_k]$ for BEAM31 elements.
- *Step 2*: matrices and vectors $\mathbf{B}_k$ (23, 27) finite elements' flexibility $[D_k]$ (Gallagher 1975; Kalanta 1995) and yield conditions $[\Phi_k]$ are created:

$$\left[D_k\right] = \begin{vmatrix} \left[D_k\left(G_{0,k}\right)\right], & k \in K_{11}; \\ \left[D_k\left(G_{1,k}, I_{y,k}\right)\right], & k \in K_{31}, \end{vmatrix} \quad (48)$$

$$\left[\Phi_k\right] = \begin{vmatrix} \left[\Phi_k\right], & k \in K_{11}; \\ \left[\Phi_k\left(G_{0,k}, G_{1,k}\right)\right], & k \in K_{31}, \end{vmatrix} \quad (49)$$

$$\mathbf{B}_k = \begin{vmatrix} \mathbf{B}_k\left(G_{0,k}, G_{1,k}\right), & k \in K_{11}; \\ \mathbf{B}_k, & k \in K_{31}, \end{vmatrix} \quad (50)$$

$$k \in K,$$

from which quasidiagonal matrices $[D]$ and the structure's configuration matrix $[B]$ are constructed. The derivative values' matrices of the mathematical model (1)–(9): $[K]$, $[H]$, $[Q]$, $[Z]$, $[Y]$ are calculated.

- *Step 3*: with reference to formulas (15, 16), the displacement vector $\mathbf{u}_e$ and internal forces' vector $\mathbf{S}_e$ for the elastic solution of the stress-strain state of the structure are found.
- *Step 4*: the routine P8 solves one iteration of nonlinear mathematical programming optimization problem (1)–(9). If the procedure of solving is successful (i.e. optimal solution is found), we have a new leading geometry vector $\mathbf{G}_0^*$ of cross-sections and a new vector $\boldsymbol{\lambda}$ of plastic multipliers.

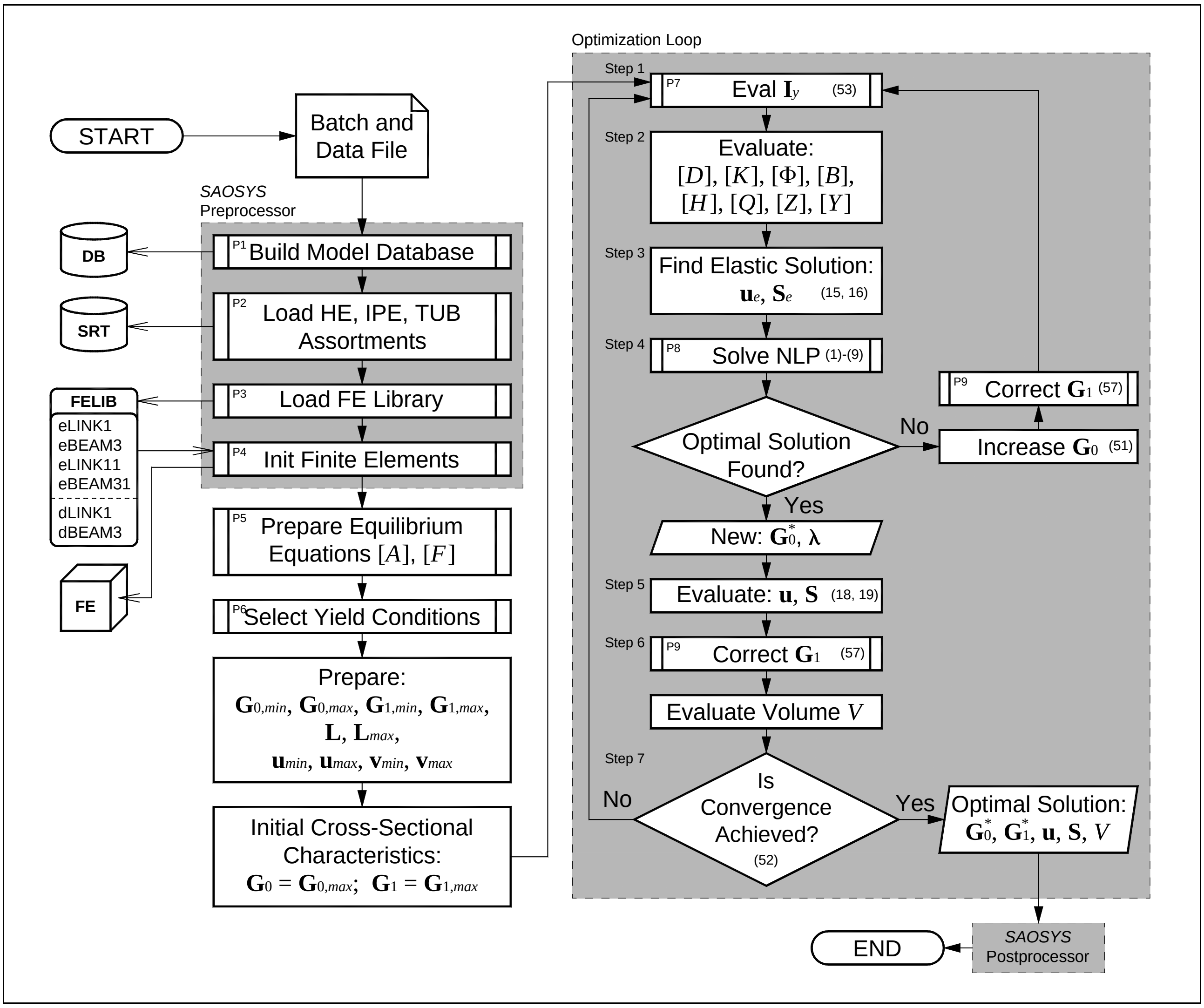

Fig. 9. Design algorithm *SAOSYS-EPSOptim* for elastic-plastic steel structures

If the solution failed (i.e. the admissible point and optimal solution are not found), the leading geometry vector $\mathbf{G}_0$ is increased:

$$G_{0,k} = \begin{cases} \frac{1}{2}\left(G'_{0,k} + G_{0,k}\right), & \frac{\left|G'_{0,k} - G_{0,k}\right|}{G'_{0,k}} \geq \varepsilon; \\ \xi G_{0,k}, & \text{otherwise}, \end{cases} \quad k \in K, \quad (51)$$

where $G'_{0,k} \equiv G_{0,k}^{*,\,prev}$ is the leading geometry vector of previous successful iteration; ε is the relative threshold of recurrent $G_{0,k}$ increase (10^{-3} %); ξ is the partial ratio of $G_{0,k}$ direct increase. The routine P9 corrects the driven geometry vector of cross-sections $\mathbf{G}_1$; then, we return to *Step 1*.

- *Step 5*: with reference to formulas (18, 19) the displacement vector **u** and internal forces' vector **S** of the real (total) stress-strain state are calculated.
- *Step 6*: the routine P9 performs a correction procedure of the driven geometry vector $\mathbf{G}_1$. The procedure is described in detail in Section 7.2.
- *Step 7*: with reference to geometry matrix $[G^*]$ cross-section areas $A^*_{0,k}$ and A^*_k values, the structure's volume V is calculated. This iterative process is performed until the above convergence conditions of the problem are satisfied:

$$\begin{cases} \max\left\{ \frac{\left|G'_{0,k} - G^*_{0,k}\right|}{G'_{0,k}}, k \in K \right\} \leq \varepsilon_{G_0}, \\ \frac{\left|V' - V\right|}{V'} \leq \varepsilon_V, \end{cases} \quad (52)$$

where V' is structural volume of the previous iteration; ε_{G_0}, ε_V denote convergence tolerance criteria (0,1 %) of the leading geometry of cross-sections and structural volume, respectively.

The results of problem solution verification. Based on the optimal solution $\mathbf{G}_0^*, \mathbf{G}_1^*$, **u**, **S**, V, the postprocessor of the system *SAOSYS* calculates the distribution of strength reserves in the elements' length and creates the control diagram *EYCPlot* of the elements' strength reserve. In addition, the internal forces and structural strain intensity diagrams can be created, and the numerical results of node displacements, as well as extreme deflections and internal forces of the elements can be derived.

7.1. Interpolation of the inertia moments of cross-sections

Let us assume that the admissible point of cross-section ${}^{k}\mathbf{G}$ belongs to the discrete field $\mathscr{D}_{A-W_{pl,y}}$ of geometric characteristics of assortment profiles. We have to find the moment of inertia $I_{y,k}$, depending on 3D discrete point dispersion $\mathscr{D}_{I_y-A-W_{pl,y}}$ of assortment (Fig. 10).

Three discrete points $\mathbf{p}_i = \left\{ {}^{\mathscr{D}}A_i, {}^{\mathscr{D}}W_{pl,y,i}, {}^{\mathscr{D}}I_{y,i} \right\}$, $i = 1, 2, 3$ closest to the point ${}^{k}\mathbf{G}$ (according to the shortest geometrical distance) should be found in the normalized $\|A - W_{pl,y}\|$ system of the coordinates. Linear interpolation of the inertia moment is performed for three closest points $\mathbf{p}_i$ only when the point $\mathbf{t}$ gets in the interpolation field Ω (Fig. 10), otherwise, the mean value of inertia moments of discrete points is used:

$$I_y\left(W_{pl,y}, A, [p]\right) = \begin{vmatrix} \alpha_{g1}W_{pl,y} + \alpha_{g2}A + \alpha_{g3}, & \mathbf{t} \in \Omega; \\ \frac{1}{3}\sum_{i=1}^{3} {}^{\mathscr{D}}I_{y,i}, & \mathbf{t} \notin \Omega. \end{vmatrix} \quad (53)$$

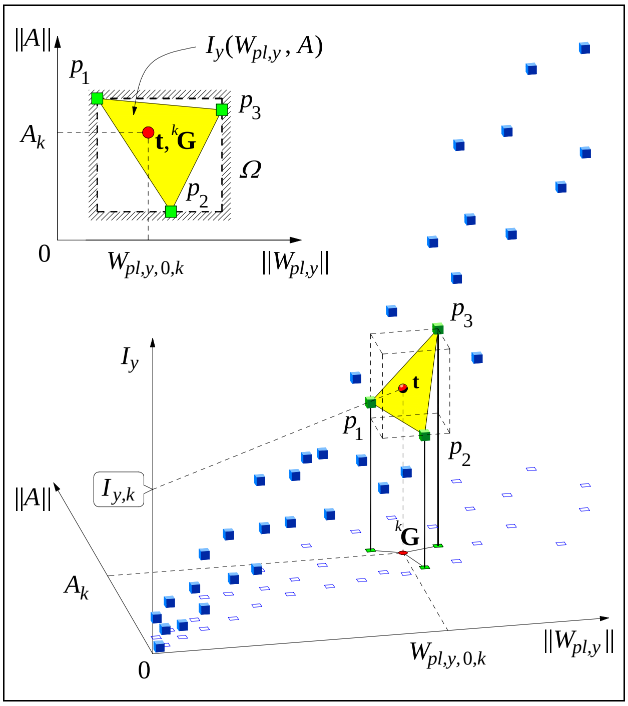

Fig. 10. Interpolation of the inertia moments of cross-sections $I_{y,k}$ of BEAM31 elements

This system of equations (created for three closest points $\mathbf{p}_i$) should be solved in order to find the linear interpolation ratios $\alpha_{g,i}$:

$$\begin{cases} \alpha_{g,i}\,{}^{\mathscr{D}}W_{pl,y,i} + \alpha_{g,i}\,{}^{\mathscr{D}}A_i + \alpha_{g,i} = {}^{\mathscr{D}}I_{y,i}, \\ i = 1, 2, 3, \end{cases} \quad (54)$$

which can be written in the matrix form as follows

$$\left[M_g\right]\boldsymbol{\alpha}_g = {}^{\mathscr{D}}\mathbf{I}_y, \quad (55)$$

and, finally, linear interpolation ratios can be derived from this formula

$$\boldsymbol{\alpha}_g \equiv \begin{Bmatrix} \alpha_{g1} \\ \alpha_{g2} \\ \alpha_{g3} \end{Bmatrix} = \left[M_g\right]^{-1} {}^{\mathscr{D}}\mathbf{I}_y. \quad (56)$$

7.2. Correction of cross-section geometry $\mathbf{G}_1$

In structural design, we introduce a concept of the set of the elements' subsets R. We optimize cross-sectional geometrical characteristics $G_{0,r}$, $G_{1,r}$ of separate subsets of elements $r \in R$. These characteristics compose a couple of vectors $\mathbf{G}_0$ and $\mathbf{G}_1$. Since we operate with the subsets of elements, the elements of the subset r are obtained by the intersection of element sets K and R_r ($K \cap R_r$). The driven geometry vector $\mathbf{G}_1$ of cross-sections is treated as the limit geometry vector, which satisfies yield conditions (2, 6) of the elements and the limits of admissible discrete fields $\mathscr{D}_{A-W_{pl,y}}$ and $\mathscr{D}_{I-A}$ of profile assortments. It can be described by the following dependencies:

$$G_{1,r} \equiv \begin{vmatrix} I_{lim,r} = I_{lim}\left(G_{0,r}, \mathbf{S}_k, L_{max,r}, \mathscr{D}_{I-A}\right), & k \in K_{11}; \\ A_{lim,r} = A_{lim}\left(G_{0,r}, \mathbf{S}_k, \mathscr{D}_{A-W_{pl,y}}\right), & k \in K_{31}, \end{vmatrix} \quad (57)$$
$$k \in K \cap R_r, \quad r \in R.$$

LINK11. We will deal with estimation of the inertia moments $I_{lim,r}$ of LINK11 elements' limit cross-sections (57), which satisfy the yield conditions (20) and discrete limits of assortments. Let us note that $A_{0,r} \equiv G_{0,r}$. For every compressed element ($N_k < 0$) of the set R_r of the elements' subsets we calculate limit buckling ratios as follows:

$$\varphi_{lim,r,k} = \frac{|N_k|}{f_{y,r}A_{0,r}}, \quad k \in K_{11} \cap R_r. \quad (58)$$

While calculating the limit buckling and slenderness ratios of this system of conditions:

$$\begin{cases} \tilde{\varphi}\left(E_r, f_{y,r}, \lambda_{lim,r,k}\right) = \varphi_{lim,r,k}, \\ \lambda_{lim,r,k} \le \lambda_{c,u}, \end{cases} \quad (59)$$

we get the limit slenderness ratios $\lambda_{lim,r,k}$ of compressed elements. Here, $\lambda_{c,u}$ is the ultimate slenderness ratio of centrally compressed element (STR 2.05.08. 2005). Finally, according to the limit slenderness ratios $\lambda_{lim,r,k}$, we get the limit inertia moment of the set R_r as follows:

$$I_{lim,r} = \max\left\{ A_{0,r}\left(\frac{L_{max,r}}{\lambda_{lim,r,k}}\right)^2, k \in K_{11} \cap R_r \right\}, \quad r \in R. \quad (60)$$

It is additionally checked and corrected. Then, in the cases of the elements under compression or tension, the limit conditions of discrete admissible fields of profile assortments must be satisfied

$$^{\mathscr{D}}I_{min,r}\left(A_{0,r}\right) \le I_{lim,r} \le {}^{\mathscr{D}}I_{max,r}\left(A_{0,r}\right). \quad (61)$$

Since $\tilde{\varphi}(\cdot)$ is a gradually decreasing piecewise function, the binary search algorithm realizes a numerical solution of the system of conditions (59) (Fig. 11). This algorithm can be implemented by using *MATLAB-SAOSYS* routine

```
x = BinFunArgValSearch(hFun, y, vInt, tol),
```

where `x` is the function argument value found ($\lambda_{lim,r,k}$); `hFun` is the function handle; `y` is the function output ($\varphi_{lim,r,k}$); `vInt` is the search interval vector $\{0; \lambda_{c,u}\}$; `tol` is the search tolerance (convergence tolerance criterion).

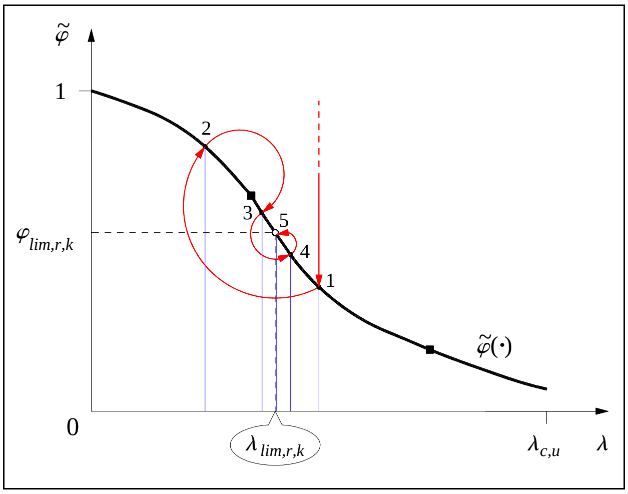

Fig. 11. Binary search algorithm for defining limit slenderness ratio $\lambda_{lim,r,k}$ of LINK11 element

Discrete limits $^{\mathscr{D}}I_{min,r}(\cdot)$, $^{\mathscr{D}}I_{max,r}(\cdot)$ (61) can be derived with reference to binary-bared search, which can be performed by this function

```
vp = BinBarSearch(vD, x, b),
```

where `vp` is the index vector of discrete points found in the bar; `vD` is the vector ($^{\mathscr{D}}\mathbf{A}_{HE \vee IPE \vee RHS}$) of discrete values arranged in the increasing order; `x` is the real value ($A_{0,r}$); `b` is the width of the search bar. According to the vector `vp`, we get discrete limits and return the non-admissible points $^{r}\mathbf{G} = \{A_{0,r}, I_{lim,r}\}$ (Fig.12, points *1* and *3*) to the admissible field (points *4* and *5*).

BEAM31. We will deal with estimation of limit cross-section areas $A_{lim,r}$ (57) of the BEAM31 elements which satisfy the yield conditions (25, 32) and discrete limits of assortments. Let us note that $W_{pl,y,0,r} \equiv G_{0,r}$. Similarly, for every element $k \in K_{31} \cap R_r$, we calculate the

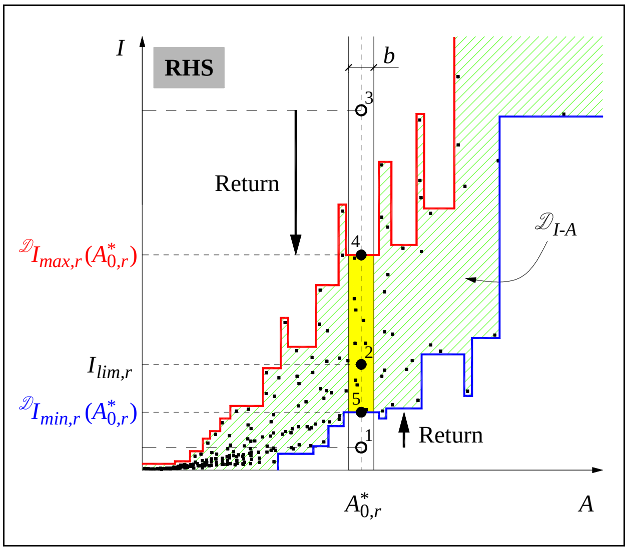

Fig. 12. Binary-bared search: return to the admissible field

limit cross-section areas $A_{lim,r,k}$, which compose the vector $\mathbf{A}_{lim,r}$ of the limit cross-section areas:

$$\mathbf{A}_{lim,r} = \left\{A_{lim,r,k}, \quad k \in K_{31} \cap R_r\right\}, \quad (62)$$

where

$$A_{lim,r,k} = \frac{\left|N_k\right|}{f_{y,r} - \xi\dfrac{\left|M_{k,i}\right|}{W_{pl,y,0,r}}}, \quad (63)$$

$$i = 1, 2, \langle 3, ext\rangle_{C^2}.$$

Finally, we get the limit cross-section area of the set of the elements' subsets R_r as follows:

$$A_{lim,r} = \max \mathbf{A}_{lim,r}, \quad r \in R. \quad (64)$$

It is also additionally checked and corrected. Then, the following limit conditions of discrete admissible fields of assortments must be satisfied

$$^{\mathscr{D}}A_{min,r}\left(W_{pl,y,0,r}\right) \le A_{lim,r} \le {}^{\mathscr{D}}A_{max,r}\left(W_{pl,y,0,r}\right). \quad (65)$$

8. *SAOSYS* system of structural modelling, analysis and design

JWM SAOSYS Toolbox v0.42 (Structural Analysis and Optimization System) is an experimental prototype of toolbox for *MathWorks MATLAB* software environment, intended for numerical research, embracing a set of data, functions, objects and scripts used in the analysis and optimal design of steel structures by finite element method. The *MATLAB* environment selected is easily used, as well as having numerous functional and technological facilities. Combined with the key module *Optimization Toolbox*, used for solving optimization problems, it became an effective tool for experimental system design.

SAOSYS system architecture is based on *User mode* pre-post-processor control system and private active *Kernel mode* block, which embraces system topology and its components (Fig. 13). Processors give *SAOSYS* control commands, functions and scripts to user for modelling and solving the problem and for interpreting textual and graphical data. Further, the development of the experimental system will be aimed at realizing the control of all processor commands.

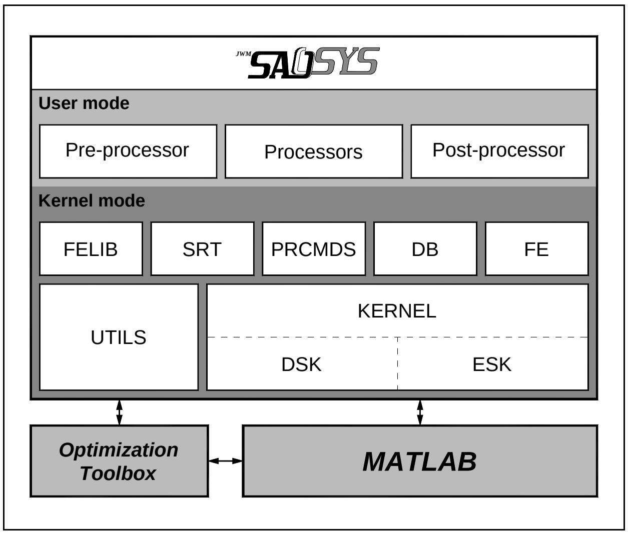

Fig. 13. The architecture of *JWM SAOSYS Toolbox v0.42*

The system's kernel is composed of the library FELIB of object-oriented finite elements; the database SRT of steel profile assortments; the processor modules PRCMDS for problem solving; the database DB of the problem and the structure; the assemblage FE of finite element objects composing the structure; the collection UTILS of additional neutral functions; and the system's functional core KERNEL composed of two subkernels DSK and ESK, working with displacement and equilibrium finite elements' formulations, respectively, for particular problem solving.

Pre-processing module. This system module is intended for the preparation of *SAOSYS* environment and structural modelling by finite element method, including the creation of load cases. The system pre-processor functions according to the deterministic finite state machine (DFSM) principle for parsing the formatted strings with comma-separated values (CSV) (Aho *et al.* 1986; Hopcroft *et al.* 2001; Bucknall 2001). There are two command groups of specifying and action. Specifying commands define the operations, which further are performed by action commands. Command arguments are characterized by the required (noted as `arg`) and default (noted as `arg`) values (Table 2).

For the discussed structural discrete model, we perform problem modeling by *SAOSYS* pre-processor commands, i.e. create Batch and Data File (BDF). The activated pre-processor (using BDF) creates *SAOSYS* structural model database DB, prepares steel profile assortment database SRT, finite elements' library FELIB, and, according to the data in DB, object-oriented finite elements' model assemblage FE of the structure and initializes the finite elements. Further, pre-processor leaves the work to the selected processor modules.

Table 2. *SAOSYS* pre-processor commands

`/TITLE``, title`
Defines the main problem title.

`/EFORM``, ef`
Chooses a formulation of system finite elements: displacement or equilibrium finite elements.

`NDOF``, ndof`
Chooses the problem type: truss or plane problem.

`MP``, id, optpars`
Defines a list of physical properties of the material.

`R``, id, optpars`
Defines a group of the element's attributes.

`N``, id, x, y, z`
Defines a discrete model node.

`E``, id, nd1, nd2, ..., ndN`
Defines an element by node connectivity.

`TYPE``, type`
`MAT``, mid`
`REAL``, rid`
Specifying commands of the elements' types, materials and element attributes' pointers.

`ER``, eid, [eNd1 eNd2 ... eNdM]`
Declares the releases of the elements' nodes.

`LOAD``, id, name`
Defines a load case.

`D``, [nd1 nd2 ... ndM], dof`
Defines DOF constraints at the nodes.

`F``, [nd1 ...], Fx, Fy, Fz, Mx, My, Mz`
`SFE``, [eid1 ...], axis, p1, p2`
Specifies concentrated and distributed loads.

Processing modules. In the present system version (v0.42), the following PRCMDS processing modules are realized for problem solving: *StatAn* is the static structure's analysis; *EPSOptim* is the optimal elastic-plastic steel structure's design (analyzed in this paper); *TrussDPD* is the direct probability design of optimal steel trusses (Jankovski and Atkočiūnas 2008).

Every processing module has the main subroutine, which prepares a supportive environment of algorithm and performs problem solving. When the solving procedure is completed, the main subroutine copies all the data from the local memory stack (the routine internal workspace) and pastes it into the global *MATLAB base workspace* memory for further post-processing interpretation.

Post-processing module. This module is intended for output and interpretation of textual and graphical results. The following graphic functions are noteworthy: `NPlot()` displays nodes; `EPlot()` is used for deformed and non-deformed structural schemas of displays; `SPlot()` creates diagrams of internal forces; `EYCPlot()` creates reserve diagrams of the elements' strength; `UPlot()` creates the deformed schema of the structure, according to intensity values of displacement $\mathbf{u}_X$, $\mathbf{u}_Y$ and $\mathbf{u}_\Sigma$.

Model database. All information about the problem and finite elements model of the structure is placed in the *SAOSYS* database DB of structural model (Fig. 14). It allows us to create a usable structure of the initial data, serving as a basis for finite element model creation. The database consists of the following tables: materials of structural elements MAT; attributes of structural element groups REAL; the discrete structure model nodes NODE; finite elements of the structure ELEM; groups of external loads LOAD; load cases LVAR. The problem is additionally defined by the finite elements' formulation EFORM, degree of freedom of discrete model node NDOF, problem title TITLE and other parameters.

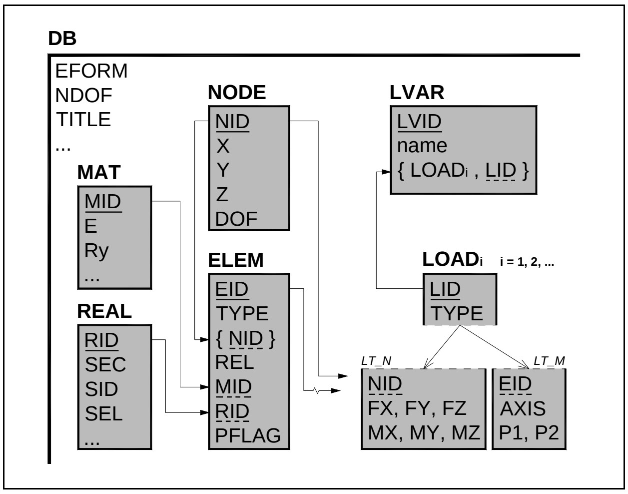

Fig. 14. *SAOSYS* database DB of structural model

Database of steel assortments. The design of steel structures is dependent on databases of profile assortments. Relational Database Management System (RDBMS) and its internal integration to *SAOSYS* system are refused. Instead, for simplicity, the plain text files, binary files and *MATLAB* structural principles are used. The system operates with HE, IPE and RHS steel profile assortments. We will discuss the creation of IPE assortment prior to its use in design.

Steel profile characteristics are written in a not formatted plain text file *IPE.sec* (Fig. 15). The assortment file has two directives: `#srt` is the assortment name; `#par` denotes the labels of column data. Profile names and geometric characteristics are given in the rows below.

```
; IPE Steel Sortiment

#srt  IPE
#par  name A Iy Wy Wply Iz

IPE_A80     6.38   64.38  16.51  18.98   6.85
IPE_80      7.64   80.14  20.03  23.22   8.49
IPE_A100    8.78  141.20  28.81  32.98  13.12
IPE_100    10.32  171.00  34.20  39.41  15.92
IPE_A120   11.03  257.40  43.77  49.87  22.39
IPE_120    13.21  317.80  52.96  60.73  27.67
IPE_A140   13.39  434.90  63.30  71.60  36.42
IPE_140    16.43  541.20  77.32  88.34  44.92
...
```

Fig. 15. A fragment of profile assortment file *IPE.sec*

SAOSYS function `Sortiment()` compiles text file *IPE.sec* and creates binary assortment file *IPE.seb*, which consists of a header and data segment. Further, *SAOSYS* works only with binary files. Therefore, assortment data reading into *MATLAB* workspace memory is fast. Function `Sortiment()` works in the reading mode. It reads the selected binary assortment and returns data structure `IPE`, which is defined by the assortment type `TYPE`; name `srtName`; names of profile characteristics' `parName`; and the number of profiles and arrays of characteristics' values. All profile assortments are integrated into SRT database, i.e. assortment table (Fig. 16), which is later used by *SAOSYS* design algorithms.

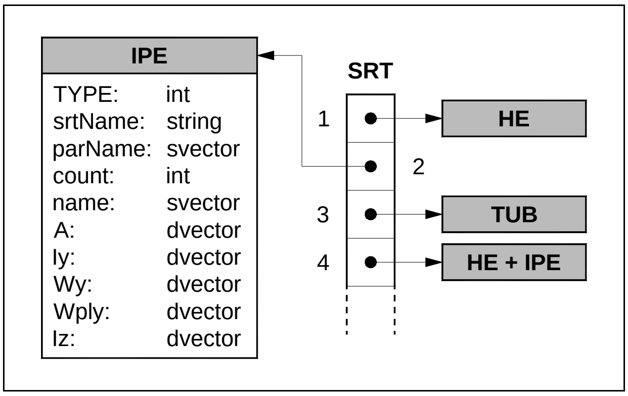

Fig. 16. The structure of steel profile assortments

9. Object-oriented finite elements and pseudo-polymorphism

The system of structural modelling, analysis and design *SAOSYS* operates with object-oriented finite elements. The main choice of object-oriented programming (OOP) was determined by the concepts of encapsulation and polymorphism (Gamm *et al.* 1995; Riel 1996; Eckel 2000). The library FELIB of the system's finite elements consists of the following classes of finite elements: elastic elements LINK1, BEAM3 and elastic-plastic elements LINK11, BEAM31. Finite element class is a collection of properties (variables) and methods (functions) working with these variables. New finite elements can be integrated into *SAOSYS* for new problem solving.

An element class constructor creates the finite element object in *MATLAB* environment memory. Object characteristics and pointers to methods (in nested function forms) are placed inside the object, i.e. in its data field. Standard *MATLAB 7.0* object control methods `subsasgn()` and `subsref()` are overloaded, and this allowed us to create a compact syntax form, similar to that used in *C++* programming language:

```
obj.member = expr                                (66)
obj.member(expr [, expr]) = expr                 (67)
obj.member{expr} = expr                          (68)
designator = obj.member                          (69)
designator = obj.member(expr [, expr])           (70)
designator = obj.member{expr}                    (71)
designator = obj.Member([exprlist])              (72)
```

The *MATLAB* functions `get()` and `set()` are generally intended for object manipulation (Krysl 2005; Register 2007). Having disposed of these inconvenient functions, we achieved a marked improvement in the program's text clarity and application simplicity. Direct assignment of expression `expr` result to object member `obj.member` is performed by (66); the assignment of the results to arrays and cell array members is performed by (67, 68); direct assignment of object member value is performed by (69); assignments of array and cell array member values are performed by (70, 71). Syntactic collision of calls to object methods (72) and assignment of object submember (70) is solved by using the lists of method names placed in the classes. These lists separate object members-data from object members-methods.

A polymorphism concept is used to perform flexible operations with finite elements. *MATLAB 7.0* system provides limited OOP facilities, not supporting polymorphism. Therefore, pseudo-polymorphism is implemented in *SAOSYS* system. In this concept, the basic class of finite element represents the cell array FE of structural elements' assemblage. The cell array FE maintains a concept of pseudo-virtual methods (Fig. 17) in the derived classes of elements. The base class of finite elements FE is more oriented to beam elements and has the following properties: dimension degree SPACE; type identificator TYPE; element number ID; the number of the members SNUM of the vector of the element's internal forces $\mathbf{S}_k$; the number of the element's displacement vector $\mathbf{u}_k$ members UNUM; the number of the yield conditions PHINUM of the element nodes; the element's material identificator MID; the element group identificator RID; the element length L, direction cosines vector `vDir`; etc. The base class of finite element implements such pseudo-virtual methods: element initialization `Init()`; initiator-selector of the element's yield conditions `InitYC()`; the getting methods of the transformation matrix $[T_k]$ `GetTk()`, the matrix of coefficients of the structure's equilibrium equations $[A_k]$ `GetAk()`, flexibility matrix $[D_k]$ `GetDk()`, yield conditions $[\Phi_k]$ and $\mathbf{B}_k$ `GetYC()`; the method of deformed element and evaluation of displacement intensity distribution `EvalDefU()`; the printing method of the element's internal data `EPrint()`; the method of the deformed and undeformed element plotting `EPlot()`; methods of the diagram output of the element's internal forces `SPlot()` and strength reserve `EYCPlot()`. The classes `eLink11` and `eBeam31` of the derived elements are complemented with additional characteristics and overloaded methods, including private helper routines.

First, we choose the formulation of the finite element method (equilibrium or displacement finite elements). Then, the *SAOSYS* system calls finite element class constructors (`eBeam31()`, `eLink11()`) in turn and prepares the library FELIB (the table of objects of the finite element types) (Fig. 18). In *MATLAB* environment, the array FELIB represents object samples.

While parsing BDF, the *SAOSYS* pre-processor places finite elements in the element assemblage array FE. Here we can indicate three new element placement steps: 1. selection – queried finite element object selection from FELIB; 2. copying – queried element object copying into FE; 3. initialization – sample element initialization method `Init()` call, which initializes finite element properties and method pointers.

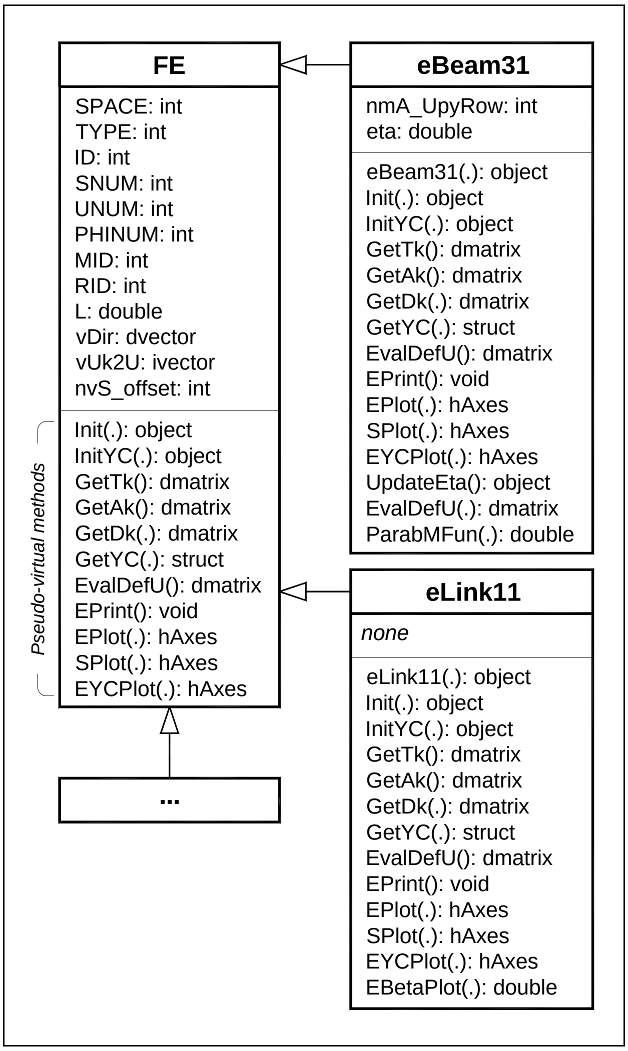

Fig. 17. UML diagram of finite element classes

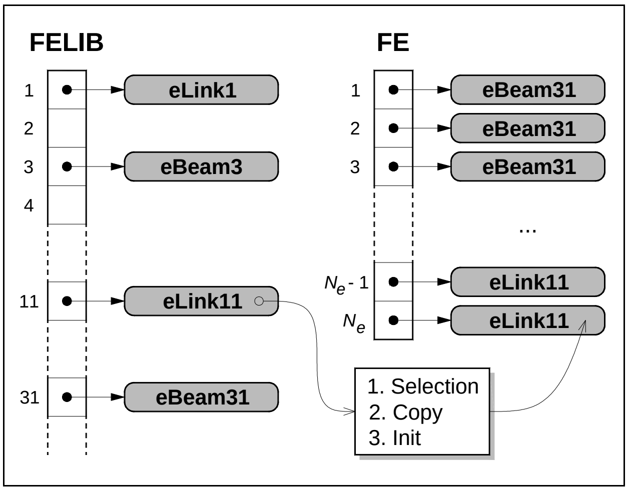

Fig. 18. Library-table of the objects of the finite elements FELIB and structural elements' assemblage FE

The array FE, validated by pseudo-virtual methods, prompts to the realization of objects independent of an object type. For example, to derive the matrix of coefficients of the structure's equilibrium equations of k-th finite element, we need the prompt `mAk = FE{k}.GetAk()`. *MATLAB 7.0* environment does not allow us to prompt directly to the memory addresses of the data fields (because it does not maintain a pointer data type). Therefore, the values of the variables are updated by using reinitialization principle, which lowers the efficiency of *MATLAB* environment, especially, in the case of objects. Therefore, contrary to other high-level programming languages (*C++*, *Delphi*, *Java*), it is necessary to update the object by reassigning `FE{k} = FE{k}.Init(id)`.

10. Graphical representation of displacement intensity of the deformed element

The optimization problem (1)–(9) is solved. Then, according to (18), the real displacement vector $\mathbf{u}$ of the structural nodes is computed. *SAOSYS* post-processor routine `UPlot()` allows us to create the deformed schema, which shows the distribution of displacements' ($\mathbf{u}_X$, $\mathbf{u}_Y$, $\mathbf{u}_\Sigma$) intensities in the structure.

Let us discuss a procedure of the finite element BEAM31 presentation (Fig. 19). The vector $\mathbf{u}_k$ of the finite element nodes' displacement from the vector $\mathbf{u}$ is selected. Invoking the function-matrix $[H_k(x)]$ for producing the element displacements' interpolation form, we created the element displacements' interpolation vector function $\mathbf{u}_k(x) = \{u_{k,x}(x), u_{k,y}(x)\}$ in the system of the local coordinates (LCS) $x1y$.

We divide an element into N equal sections (Fig. 19 a). Then, the coordinate vectors $\mathbf{P}_i \equiv \{P_{i,X}, P_{i,Y}\}$ in the global system of the coordinates (GCS) are calculated as follows:

$$\mathbf{P}_i = \mathbf{J}_1 + [T_k']^T \left(\begin{vmatrix} x_i \\ 0 \end{vmatrix} + s\left[H_k(x_i)\right][T_k]\mathbf{u}_k \right), \qquad (73)$$

$$x_i = \frac{l_k}{N} i, \quad i = 0, 1, 2, ..., N, \qquad (74)$$

where $[T_k]$ is the transformation matrix of node displacements of the element GCS→LCS; s is the scale ratio; $[T_k']$ is the transformation matrix of the coordinates (2×2); $\mathbf{J}_1$ is the vector of the element's first node coordinates in GCS.

Let the element be described by the thickness t (Fig. 19 c, d). Then, this condition (the sum of geometric conditions of the vectors (75)) is valid for calculating the vectors of the coordinates $\mathbf{L}_i$ and $\mathbf{R}_i$ of the element's multiline nodes

$$\frac{t}{2}\boldsymbol{\tau}_i + \gamma \mathbf{n}_i = \xi \mathbf{r}_i, \qquad (75)$$

$$\mathbf{n}_i = \frac{\mathbf{P}_{i+1} - \mathbf{P}_i}{\|\mathbf{P}_{i+1} - \mathbf{P}_i\|}; \quad \boldsymbol{\tau}_i = \{-n_{i,Y}, n_{i,X}\}; \quad \mathbf{r}_i = \boldsymbol{\tau}_i + \boldsymbol{\tau}_{i-1}. \qquad (76)$$

We solve the system of equations (75) in respect of the unknown ratios γ and ξ. Finally, the coordinates of multiline nodes are calculated by the formulas given below:

$$\mathbf{L}_i = \mathbf{P}_i + \xi \mathbf{r}_i, \quad \mathbf{R}_i = \mathbf{P}_i - \xi \mathbf{r}_i, \qquad (77)$$

$$\xi = \frac{t}{2} \cdot \frac{1}{1 + \mathbf{n}_i^T \mathbf{n}_{i-1}}. \qquad (78)$$

The sections of the deformed element $\mathbf{R}_{i-1}$-$\mathbf{R}_i$-$\mathbf{L}_i$-$\mathbf{L}_{i-1}$ are covered with quadrangles (Wright and Lipchak 2004), the nodes' colour vectors of which $\mathbf{c}_j \equiv \{R_j, G_j, B_j\} = \mathbf{c}_j(u_{j,X \vee Y \vee \Sigma})$ are displacement intensity functions of nodes $\mathbf{P}_i$ in respect of all displacements of structural elements.

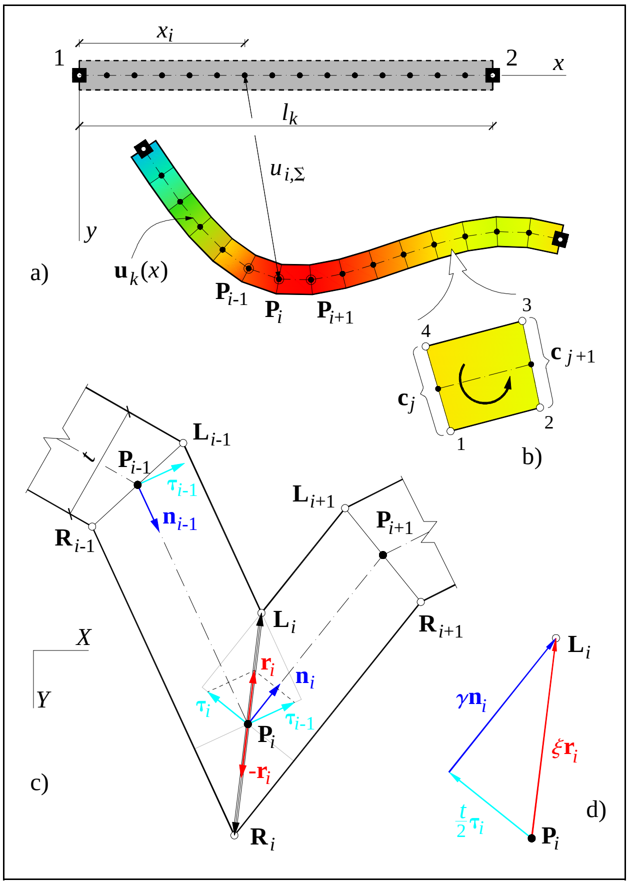

Fig. 19. Displacements and strains of the beam element: a) deformed element BEAM31; b) sequence of element surface colouring; c) creation of deformed element multiline; d) geometric condition of the sum of vectors

11. A numerical example

Design structure. Industrial building frame subjected to simple loading is designed at the elastic-plastic stage (Fig. 21). The frames are placed along the building at the interval of $L = 6{,}0$ m. The element's material is steel S275: $E = 210$ GPa, $R_y = 275$ MPa. The frame is modelled by equilibrium finite elements and consists of 19 nodes, 30 finite elements (13 of the BEAM31 type, 17 of the LINK11 type), and 14 design parameters R_{1-14} (i.e. element cross-sections). The first-floor columns R_1–R_3 are designed from HE type profiles. The second-floor columns R_4 are designed from HE or IPE type profiles (the alternative design is performed for cross-section type). The girders R_5–R_6 are designed from IPE type profiles. The truss top chord R_7 is designed from IPE type profiles, and the bottom chord and the grid R_8–R_{14} – from RHS profiles. Structural loads are as follows: $s_{11} = s_{12} = s_2 = 12{,}480$ kN/m; $w_{11} = 1{,}123$ kN/m; $w_{12} = 0{,}562$ kN/m; $w_{21} = 1{,}393$ kN/m; $w_{22} = 0{,}696$ kN/m; $q = 23{,}400$ kN/m. The own weight of the structure is not evaluated. The stiffness conditions of the structure are as follows: for nodes – $[|u_{4x}|, |u_{7x}|] \le 0{,}050$ m, $u_{16y} \le 0{,}055$ m; for elements (deflections) – $[v_{max,6}, v_{max,7}] \le 0{,}030$ m.

Batch and Data File. The preparation of the design system, structural parameterization and sample fragment of modelling commands in pre-processor mode are contained in the file *Frame.m* of the form given in (Fig. 20). It is a plain text *MATLAB* script file, beginning with the `HEADER` consisting of *SAOSYS* environment preparation macros. Further, ordinary *MATLAB* computations (calculations of node coordinates, etc.) can be performed. The declared user-named variables (`b1`, `q0`) are the structural parameters. They are directly applied to *SAOSYS* pre-processor's command fragment, which is bracketed by "`%{ … %}`" symbols (the *MATLAB* block comment).

The following parameters are defined in the segment of *SAOSYS* pre-processor commands: the problem title `/TITLE`; finite element formulation `/EFORM` (`EF_E` denotes equilibrium finite elements); the degree of freedom of the node `NDOF` (`ND_PLANE` is the 2D problem with three freedom degrees of the node); element materials `MP` (here, elasticity modulus `E` and the tensile steel strength depending on the yield stress `Ry` should be defined); finite elements' sets `R` (here, `sec` is the cross-section type, `sid` is the default cross-section number in assortment and `sel` is the cross-section selection flag). Further, the nodes' coordinates `N` of the structure's model are defined. Finite element definition (`E` command) is performed only after specifying the pointers to the type of the element `TYPE`, material `MAT` and the element's attribute group `REAL`. Then, boundary conditions of nodes (supports) `D`, and the element's node releases (hinges) `ER` should be defined. Finally, load cases `LOAD` are created, where `F` and `SFE` commands are assigned the values of concentrated and distributed loads, respectively. At the end of the file, a prompt to *SAOSYS* processor module routine is written: `EPSOptim()` is the elastic-plastic structure's optimization with stiffness constraints. Then, the BDF can be executed.

The results obtained. Structural design was performed by using an iterative procedure. In general, eleven iterations were made (Fig. 22). As a result, optimal theoretical cross-sections were found. The profiles closest to them are presented in the table (Table 3). While using the alternative design, it is advisable to choose the frame's second-floor column $E\{4, 5\}$ cross-sections from the IPE type profiles, while, the first-floor column $E\{1, 2, 3\}$ cross-sections are chosen from HE type profiles. The volume of the designed structure is $V = 6{,}897 \cdot 10^{-2}$ m^3. In the future, we intend to realize discrete optimization of structures, which is required for correcting the final findings of the optimal discrete solution.

The diagram of strength reserve of structural elements (Fig. 23) shows the location of plastic hinges. The elements $E\{5, 10, 11, 16, 21, 22, 24, 25, 29, 30\}$ are designed under strength reserve state (i.e. plastic flow, or stability loss in the case of LINK11 is observed). Intensity diagrams of displacements $\boldsymbol{u_x}$ and $\boldsymbol{u_y}$ (Fig. 24, Fig. 25) show that stiffness requirements for the nodes and elements of the structure are satisfied.

```
HEADER                  % SAOSYS header macros

b1 = 12.0;              % Define variables as
q0 = 3.0*1.3*L;         %  model parameters.
...                     % Other MATLAB evals

%{                      % Batch segment start
/TITLE,'Frame 2008'
/EFORM,EF_E
NDOF,ND_PLANE

; Materials and gruops of elements
MP,1, E=210e6, Ry=275e3
R,1,  sec=ST_HE,     sid=1, sel=1
R,2,  sec=ST_HE,     sid=1, sel=1
R,3,  sec=ST_HE,     sid=1, sel=1
R,4,  sec=ST_HE_IPE, sid=1, sel=1
...
R,14, sec=ST_TUB,    sid=1, sel=1

N,1                                 ; Nodes
N,2, b1
N,3, b1+b2
...
N,17, b1/12 + 3*b1/6, -h1+h4
N,18, b1/12 + 4*b1/6, -h1+h4
N,19, b1/12 + 5*b1/6, -h1+h4

TYPE,BEAM31                         ; Elements
REAL,1
E,1,1,4
REAL,2
E,2,2,5
...
TYPE,LINK11
REAL,9
E,19,14,4
E,30,19,5
...
REAL,14
E,24,16,11
E,25,17,11

D,[1 2 3],UX+UY+URZ                 ; BndrCnds
ER,1,2
ER,13,2

LOAD,,'Load Case 1'                 ; Loads
SFE,1,LY,w11
SFE,3,LY,w12
SFE,[8 9 10],LY,s11 + g0
SFE,6,LY,q0
...
%}                      % Batch segment end

EPSOptim(mfilename)     % Call PRCMD routine
```

Here: *MATLAB* code space;
MATLAB + *SAOSYS* pre-processor space.

Fig. 20. Sample initial data and a batch file (BDF)

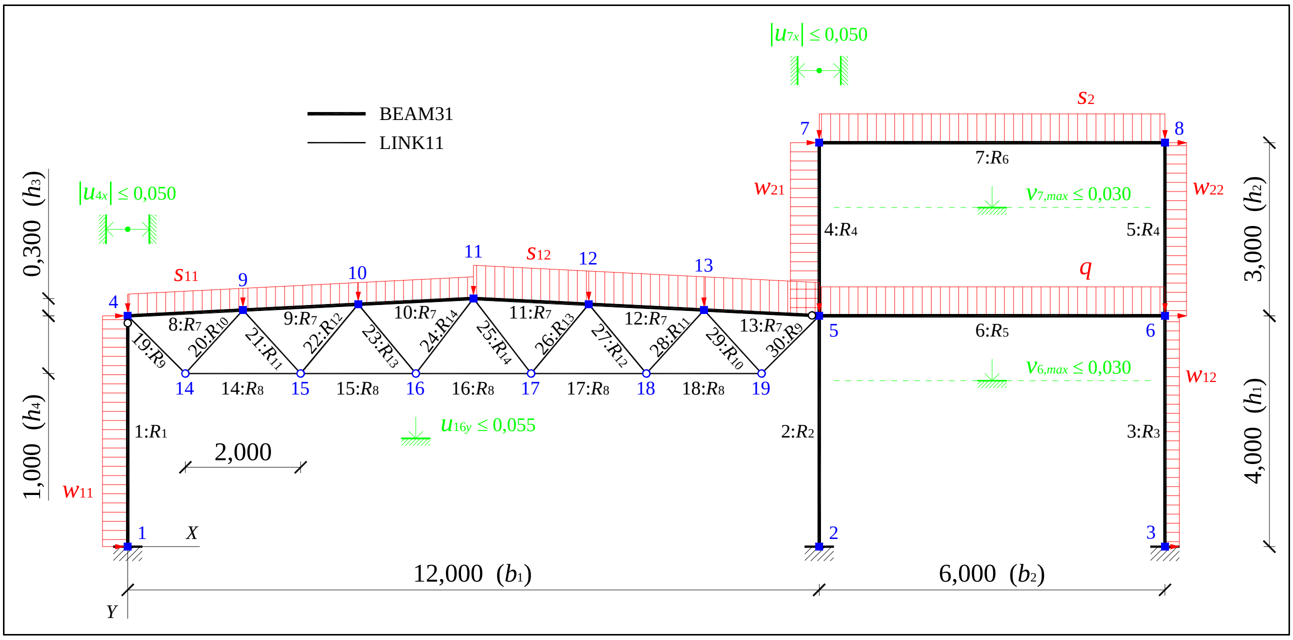

Fig. 21. Structural model discretized by LINK11 and BEAM31 finite elements

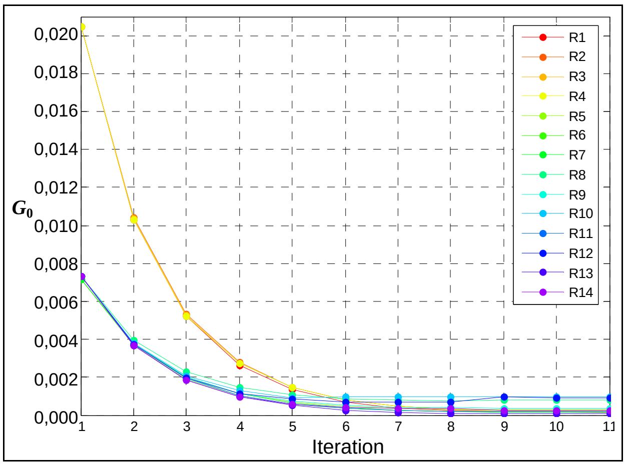

Fig. 22. Variation of G_0 in iterative calculations

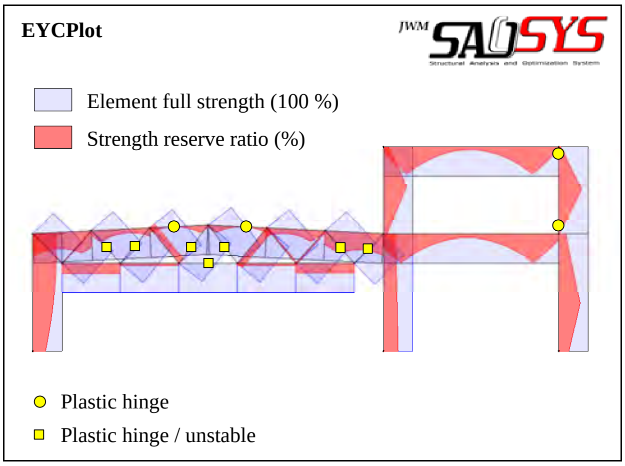

Fig. 23. The structure elements strength reserve diagram

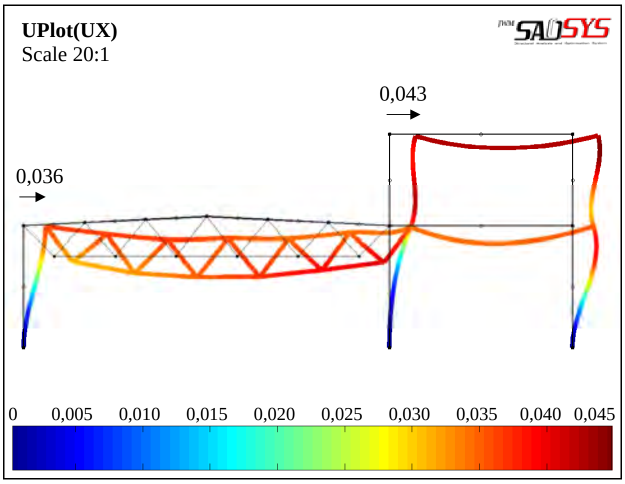

Fig. 24. Diagram $\mathbf{u}_X$ of horizontal displacements [m]

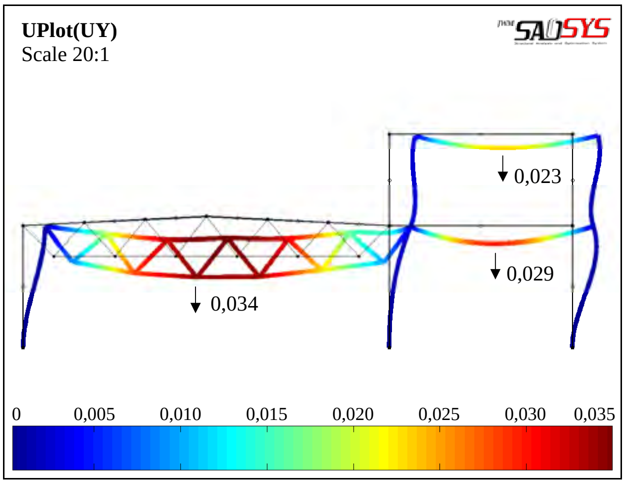

Fig. 25. Diagram $\mathbf{u}_Y$ of vertical displacements [m]

Table 3. Calculated optimal theoretical cross-sections (*) an the profiles closest to these cross-sections

	Profile	$G_{0,k}$ $W_{pl,y,0,k}\cdot 10^{-5}$, m^3	$G_{1,k}$ $A_k\cdot 10^{-3}$, m^2
R_1:	*	**5,836**	**1,560**
	HE 100AA	5,836	1,560
R_2:	*	**11,983**	**1,560**
	HE 100AA	5,836	1,560
	HE 120AA	8,412	1,855
	HE 100A	8,301	2,124
R_3:	*	**12,282**	**1,560**
	HE 100AA	5,836	1,560
	HE 120AA	8,412	1,855
	HE 100A	8,301	2,124
R_4:	*	**14,109**	**0,724**
	IPE 80	2,322	0,764
	IPE A100	3,298	0,878
	IPE A80	1,898	0,638
R_5:	*	**21,671**	**2,347**
	IPE A200	18,170	2,347
	IPE 180	16,640	2,395
	IPE O180	18,910	2,710
R_6:	*	**11,972**	**1,103**
	IPE A120	4,987	1,103
	IPE 100	3,941	1,032
	IPE A140	7,160	1,339
R_7:	*	**8,136**	**0,936**
	IPE 100	3,941	1,032
	IPE A100	3,298	0,878
	IPE A120	4,987	1,103

	Profile	$A_{0,k}\cdot 10^{-4}$, m^2	$I_k\cdot 10^{-8}$, m^4
R_8:	*	**7,171**	**6,406**
	RHS 30×70×4	7,360	10,249
	RHS 40×70×3.5	7,210	16,371
	RHS 25×60×5	7,500	6,406
R_9:	*	**3,622**	**0,808**
	RHS 20×45×3	3,540	2,108
	RHS 20×40×3.5	3,710	1,889
	RHS 25×35×3.5	3,710	2,900
R_{10}:	*	**9,403**	**14,085**
	RHS 30×70×5	9,000	11,750
	RHS 40×60×5	9,000	20,750
	RHS 40×90×4	9,760	25,609
R_{11}:	*	**1,491**	**0,0899**
	RHS 10×25×2.5	1,500	0,171
	RHS 15×20×2.5	1,500	0,385
	RHS 10×30×2	1,440	0,203
R_{12}:	*	**4,812**	**6,368**
	RHS 30×45×3.5	4,760	5,632
	RHS 25×50×3.5	4,760	3,995
	RHS 25×60×3	4,740	4,726
R_{13}:	*	**0,460**	**0,0695**
	RHS 10×15×1	0,460	0,0695
R_{14}:	*	**2,711**	**1,604**
	RHS 20×40×2.5	2,750	1,438
	RHS 25×35×2.5	2,750	2,165
	RHS 30×42×2	2,720	3,884

12. Conclusions

1. The design of elastic-plastic structures is a problem of nonlinear mathematical programming, which can be solved by the iteration method approaching the optimal solution step by step.
2. The additional nonlinear constraints-conditions in solving an optimization problem evaluate strength, stiffness and stability requirements to structural elements more accurately, enabling us to avoid densification of the finite element grid, i.e. to decrease the volume of the optimization problem and thereby the time of solution.
3. The principle of the admissible fields of geometric characteristics of the assortment profiles (the optimized leading geometry $\mathbf{G}_0$ and the controlled driven geometry $\mathbf{G}_1$) allows the design of the elements depending on the dispersion of geometric characteristics of profile sets in assortments.
4. A concept of the OOP pseudo-polymorphism realized in *MATLAB* environment allowed the authors to create such *SAOSYS* system architecture that can flexibly operate with object-oriented finite elements.
5. The *SAOSYS* system and its module *EPSOptim*, aimed at designing elastic-plastic steel structures under single loading, which were developed by the authors, do not take into account the probable stability loss of beam elements yet.

References

Aho, A.V.; Sethi, R.; Ullman, J. D. 1986. *Compilers: Principles, Techniques, and Tools.* Addison Wesley. 796 p.

Atkočiūnas, J. 1994. *Design of Elastoplastic Systems under Repeated Loading*. Vilnius: Science and Encyclopedia Publishers. 147 p. (in Russian).

Atkočiūnas, J. 1999. Mathematical models of optimization problems at shakedown, *Mechanics Research Communications*, 26(3): 319–326. doi:10.1016/S0093-6413(99)00030-0

Atkočiūnas, J.; Merkevičiūtė, D. 2003. Kuhn–Tucker conditions and load optimization problem at shakedown, in Burczynski T.; Fedelinski P.; Majchrzak, E. (Editors). *Proceedings of the 15th International Conference on Computer Methods in Mechanics.* Gliwice: Silesian Technical University.

Atkočiūnas, J.; Venskus, A. 2008. Optimal shakedown design of frames under stability conditions, in Topping, B. H. V. and Papadrakakis, M. (Editors). *Proceedings of the Ninth International Conference on Computational Structures Technology*, September 2–5, 2008, Athens, Greece, Civil-Comp Press. 12 p. ISBN 978-1-905088-22-5.

Atkočiūnas, J.; Jarmolajeva, E.; Merkevičiūtė, D. 2004. Optimal shakedown loading for circular plates, *Structural and Multidisciplinary Optimization*. International Society for Structural and Multidisciplinary Optimization (ISSMO) 27(3): 178–188. ISSN 1615-147X.

Atkočiūnas, J.; Merkevičiūtė, D.; Venskus, A. 2008. Optimal shakedown design of bar systems: Strength, stiffness and stability constraints, *Computers & Structures* 86: 1757–1768. doi:10.1016/j.compstruc.2008.01.008

Atkočiūnas, J.; Merkevičiūtė, D.; Venskus, A.; Skaržauskas, V. 2007. Nonlinear programming and optimal shakedown design of frames, *Mechanika* 2(64): 27–33. Kaunas: Technologija. ISSN 1392–1207.

Atkočiūnas, J.; Rimkus, L.; Skaržauskas, V.; Jarmolajeva, E. 2007. Optimal shakedown design of plates, *Mechanika* 5(67): 14–23. Kaunas: Technologija. ISSN 1392-1207.

Atrek, E.; Gallagher, R. G.; Ragsdell, K. M.; Zienkiewicz, O. C. 1984. *New Directions in Optimum Structural Design.* John Wiley and Sons, Chichester, New York, Brisbane, Toronto, Singapore. ISBN 5-274-00334-6.

Bazaraa, M. S.; Sherali, H. D.; Shetty C. M. 2004. *Nonlinear Programming: Theory and Algorithms.* New York: John Wiley & Sons Ltd. 853p.

Brandt, A. M. 1978. PAN IPPT, *Podstawy optymalizacji elementów konstrukcji budowlanych.* Warszawa, Państwowe wydawnictwo naukowe. 503 s.

Bucknall, J. 2001. *Tomes of Delphi: Algorithms and Data Structures.* Wordware Publishing, Inc. 525 p.

Choi, S. H.; Kim, S. E. 2002. Optimal design of steel frame using practical nonlinear inelastic analysis, *Engineering Structures* 24(9): 1189–1201. doi:10.1016/S0141-0296(02)00053-6

Čyras, A.; Borkowski, A.; Karkauskas, R. 2004. *Theory and Methods of Optimization of Rigid-Plastic Systems.* Vilnius: Technika. 354 p.

De Veubeke, B. F. 1963. *Displacement and Equilibrium Models in the Finite Element Method. Stress Analysis.* Ch. 9. John Wiley, London.

Eckel, B. 2000. *Thinking in C++.* 2nd ed. Volume One: Introduction to standard C++. Prentice Hall Inc. 842 p.

Gallagher, R. H. 1975. *Finite Element Analysis: Fundamentals.* Englewood Cliffs: Prentice-Hall Inc. 420 p.

Gamm, E.; Helm, R.; Johnson, R.; Vlisides J. 1995. *Design Patterns: Elements of Reusable Object-Oriented Software.* Addison Wesley Longman Inc. 416 p.

Giambanco, F.; Palizzolo, L.; Polizzotto, C. 1994. Optimal shakedown design of beam structures, *Structural Optimization* 8: 156–167. doi:10.1007/BF01743313

Hopcroft, J. E.; Motwani, R.; Ullman, J. D. 2001. *Introduction to Automata Theory, Languages, and Computation.* 2nd ed. Addison Wesley. 521 p.

Jankovski, V.; Atkočiūnas, J. 2008. MATLAB implementation in direct probability design of optimal steel trusses, *Mechanika* 6(74): 30–37. Kaunas: Technologija. ISSN 1392-1207.

Kalanta, S. 1995. The equilibrium finite element in computation of elastic structures, *Statyba* [Civil Engineering]. Vilnius: Technika, 1(1): 25–47 (in Russian).

Kalanta, S.; Atkočiūnas, J.; Venskus, A. 2009. Discrete optimization problems of the steel bar structures, *Engineering Structures Elsevier SCI LTD*, England, 31(Sp. Iss. SI 6): 1298-1304. ISSN 0141-0296.

Kaliszky, S.; Logo, J. 2002. Plastic behaviour and stability constraints in shakedown analysis and optimal design, *Structural and Multidisciplinary Optimization* 24: 118–124. doi:10.1007/s00158-002-0222-2

Kaneko, L.; Maier, G. 1981. Optimum design of plastic structures under displacement's constraints, *Computer Methods in Applied Mechanics and Engineering* 27(3): 369–392. doi:10.1016/0045-7825(81)90139-0

Karkauskas, R. 2007. Optimisation of geometrically non-linear elastic-plastic structures in the state prior to plastic collapse, *Journal of Civil Engineering and Management* 13(3): 183–192.

Koiter, W. T. 1960. General Theorems for elastic–plastic solids. in Sheddon, IN, Hills R, (Editors). *Progress in Solid Mechanics.* Amsterdam: North Holland.

Krysl, P. 2005. *A Pragmatic Introduction to the Finite element Method for Thermal and Stress Analysis.* With the *MATLAB* toolbox *SOFEA*. Pressure Cooker Press. 190 p.

Lloyd, Smith D. 1990. *Mathematical Programming Methods in Structural Plasticity.* International Centre for Mechanical Sciences, Courses and Lectures No 299, Springer-Verlag, Wien – New York. 435 p.

Majid, K. I. 1974. *Optimum design of structures.* London: Newnes-Butterworths.

Merkevičiūtė, D.; Atkočiūnas, J. 2006. Optimal shakedown design of metal structures under stiffness and stability constraints, *Journal of Constructional Steel Research* 62: 1270-5. doi:10.1016/j.jcsr.2006.04.020

Prager, W. 1955. Minimum weight design of plates. *De Ingenieur* 67(48): 141–142.

Raue, E.; Timmler, H.-G.; Garke, R. 2009. On the physically non-linear analysis of cyclic loaded reinforced concrete cross-sections with mathematical optimization, *Journal of Civil Engineering and Management* 15(2): 189–195. doi:10.3846/1392-3730.2009.15.189-195

Register, A. H. 2007. *A Guide to MATLAB Object-Oriented Programming.* SciTech Publishing Inc. 354 p.

Reitman, M. I. 1976. Зарубежные книги по оптимальному проектированию конструкций, *Строительная механика и расчет сооружений* 4: 62–64.

Riel, A. J. 1996. *Object-Oriented Design Heuristics.* Addison Wesley. 400 p.

Skaržauskas, V.; Merkevičiūtė, D.; Atkočiūnas, J. 2005. Optimisation des portiques dans les conditions d'adaptation avec des restrictions en déplacements, *Revue Européenne de Génie Civil* 9(4): 435–453. ISSN 1279-5119.

Šešok, D.; Belevičius, R. 2008. Global optimization of trusses with a modified genetic algorithm, *Journal of Civil Engineering and Management* 14(3): 147–154. doi:10.3846/1392-3730.2008.14.10

Tin–Loi, F. 2000. Optimum shakedown design under residual displacement constraints, *Structural and Multidisciplinary Optimization* 19(2): 130–139. doi:10.1007/s001580050093

Wilson, E. L. 2002. *Three-Dimensional Static and Dynamic Analysis of Structures.* A Physical Approach With Emphasis on Earthquake Engineering. Third Edition. Computers and Structures, Inc., University Avenue Berkeley, California. 423 p.

Wright, R.S.; Lipchak, B. 2004. *OpenGL SuperBible*, 3rd Edition. Sams. 1200 p.

ANSYS 9.0 Commands Reference and APDL Programmer's Guide. 2004. SAS IP, Inc. (El. version).

EN 1993-1-1:2005. Eurocode 3: Design of steel structures – Part 1-1: General rules and rules for buildings.

MATLAB 7.0 software documentation. 2004. MathWorks, Inc., (El. version).

STR 2.05.08. 2005. Design of steel structures. General rules. Design cod of Lithuanian Republic. Vilnius, Lithuanian Department of Environment (in Lithuanian).

Алявдин, П. В. 2005. *Предельный анализ конструкций при повторных нагружениях*: Монография. Минск: УП «Технопринт». 282 с. ISBN 985-464-727-7.

TAMPRIŲJŲ PLASTINIŲ PLIENINIŲ KONSTRUKCIJŲ ANALIZĖS IR OPTIMALAUS PROJEKTAVIMO *SAOSYS TOOLBOX* SISTEMA *MATLAB* APLINKAI

V. Jankovski, J. Atkočiūnas

S a n t r a u k a

Pasitelkus deformuojamo kūno mechanikos energinius principus ir matematinio programavimo teoriją, sudarytas minimalaus tūrio strypinių plieninių konstrukcijų, patyrusių ir plastines deformacijas, projektavimo nuo vienkartės apkrovos uždavinio patobulintas matematinis modelis. Diskretizacijai naudojami pusiausvirieji baigtiniai elementai su įrąžų interpoliavimo funkcijomis. Elementai projektuojami iš sortimentinių HE, IPE, RHS plieninių profiliuočių, atsižvelgiant į profiliuočių aibių geometrinių charakteristikų sklaidą sortimentuose. Optimalus plieninių konstrukcijų projektavimas realizuojamas autorių *MATLAB* aplinkoje sukurta eksperimentine sistema *JWM SAOSYS Toolbox v0.42*. *SAOSYS* architektūra pseudopolimorfiškai operuoja objektiškai orientuotais baigtiniais elementais. Sistemos *SAOSYS* galimybės atskleidžiamos pramoninio pastato rėmo optimalaus projektavimo su stiprumo ir standumo apribojimais pavyzdžiu. Skaičiuota atsižvelgus į mažų poslinkių prielaidas.

Reikšminiai žodžiai: optimalus projektavimas, ekstreminiai energiniai principai, matematinis programavimas, plieninės konstrukcijos, baigtinių elementų metodas, objektiškai orientuotas programavimas, *MATLAB*.

Valentinas JANKOVSKI. PhD student since 2007 at the Dept of Structural Mechanics of Vilnius Gediminas Technical University (VGTU). Research interests: engineering informatics, design of optimal elastic-plastic steel structures, graphical systems.

Juozas ATKOČIŪNAS. Doctor Habil, Professor. Head of the Dept of Structural Mechanics (VGTU). Civil engineer (1967). Dr Eng (structural mechanics, 1973). Dr Habil (mechanics, 1996). Research interests: structural and computational mechanics, applied mathematical programming, analysis and optimization of dissipative structures under repeated-variable loading.