Tensile Structures – Numerical Design Techniques

Master-Thesis

A thesis submitted in partial fulfillment

of the requirements for the degree of

Master Membrane Structures

submitted to

Anhalt University of Applied Sciences

Faculty of Architecture,

Facility Management and Geo Information

by

Peter Novýsedlák

[redacted]

[redacted]

[redacted]

Submission date: March 2017

First Tutor: Prof. Dr. Robert Off

Statement

I hereby declare that the work presented in this Master thesis, entitled

Tensile Structures – Numerical Design Techniques,

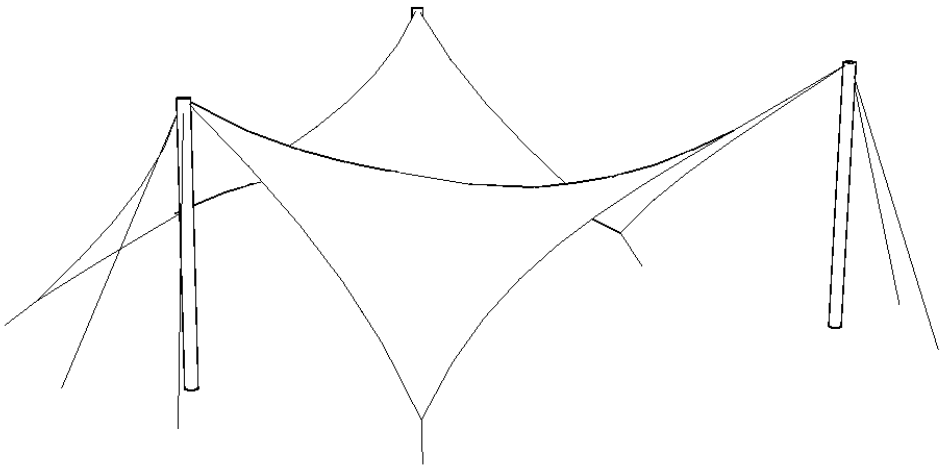
is entirely my own and that I did not use any sources or auxiliary means other than those referenced.

Dessau, March 2017

.....
Peter Novýsedlák

Contents

1 Introduction 3
2 Numerical design of tensile structures 5
2.1 Form Finding 7
2.1.1 Geometric form finding method 7
2.1.2 Mechanical form finding method 12
2.2 Analysis 14
2.2.1 Membrane element 14
2.2.2 Cable element 23
2.2.3 Wrinkling analysis 25
2.3 General solution procedure 31
2.4 Patterning 32
2.4.1 Cutting lines 33
2.4.2 Flattening 40
3 Structural optimization 49
3.1 Introduction to optimization of tensile structures 52
3.2 Basic concepts of optimization 55
3.3 Sensitivity analysis 58
3.4 Step length search 61
3.5 Unconstrained optimization methods 64
3.5.1 Optimality conditions – unconstrained minimization 64
3.5.2 Steepest descent method 65
3.5.3 Conjugate gradient method 66
3.5.4 Newton´s method 68
3.5.5 BFGS Quasi-Newton method 69
3.5.6 Comparison of methods for unconstrained minimization 71
3.6 Constrained optimization methods 72
3.6.1 Optimality conditions – constrained minimization 73
3.6.2 Penalty function method 75
3.6.3 Barrier method 76
3.6.4 Augmented Lagrange Multiplier method 78
3.6.5 Comparison of methods for constrained minimization 80
3.7 Example 81
3.7.1 Results and discussion 89
4 Summary 91
5 Literature 93



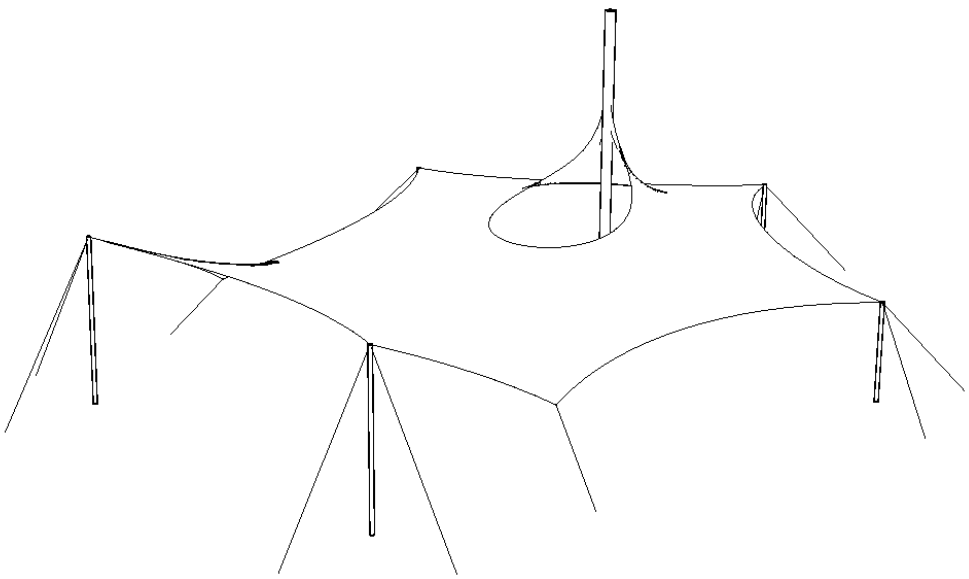
1 Introduction

There is no doubt that structural calculations, together with technical standards, contribute to safety and reliability of the structures. They also give an insight into a structural behaviour under considered circumstances. Tensile structures, which are the subject of this work, are very specific and therefore require „purpose-based“ design tools, that are usually not contained in „general-purpose“ commercial softwares. Even though the engineering office own such a software, still, high expectations and responsibility are given on the shoulders of engineer. Engineering self-confidence grows with experience and theoretical knowledge of the problem. Therefore a deep insight into how softwares work is an advantegous skill. This work offers a brief introduction into numerical design techniques and is divided in two parts.

First part of this work (Chapter 2 - Numerical design of tensile structures) is focused on numerical techniques, that are hidden behind finite element based softwares, specialized on tensile structures. During the history, many different numerical approches were developed to design and simulate their mechanical behaviour. This work collects necessary algorithms and present unified approach, based on nonlinear finite element analysis. Further description on how this method can be used in every design stage is provided in each particular chapter.

Second part (Chapter 3 - Structural optimization) deals with the possibilities of structural optimization at designing and improvement of tensile, as well as usual rigid structures. Algorithms for constrained and unconstrained optimization are presented, that are suitable also for general use. Application of these procedures opens a wide new design space, in which one is able to take into account also multidisciplinarity of enviroment. Also further improvements of structures designed with other approaches as presented here are possible. The main purpose is to achieve better structural behaviour or economical efficiency.

Content of this work reflects the author´s professional interests and should be used as an introduction for deeper individual studies from relevant literature.



2 Numerical design of tensile structures

Design of tensile structures can be divided into several phases. These phases are tightly interconnected, what results in a fact, that any change in one phase will have an influence on others. Therefore the design of tensile structures is an iterative process although its phases follow each other in linear fashion. Following design stages can be easily identified:

- architectural concept
- form finding
- structural analysis
- detailing
- cutting pattern generation

One of commonly used iterative design scheme is given on Fig. 1:

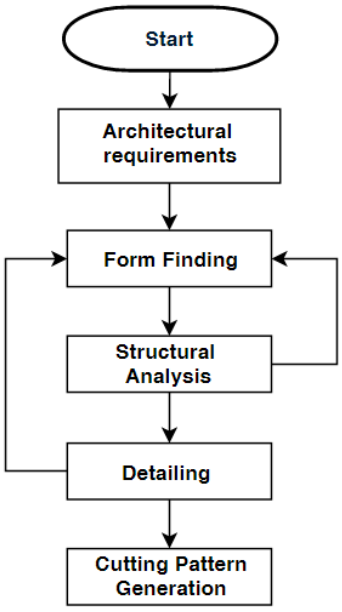


Fig. 1- General design process

Also other (even inverse) approaches are possible, for example [29], but the content of individual stages remain unchanged.

Architectural requirements
The basic definition of the structural shape is a task generally shared by clients and their architects. For structural engineer, the outputs of this stage represent the inputs for further stages. The architect collects the client´s requirements, such as height clearance, area to be covered or material type. These requirements are then collected and a sketch of a future structure with an emphasis on aesthetics is created.

Form finding
Form finding is a process to find the equilibrium state of a cable-membrane structure at a given stress level and with specified boundary conditions. At this stage, the architectural sketch is reformed into technically acceptable structure. For example the locations of secondary steel structure or prestress levels (ratios) are defined. See section 2.1.

Structural analysis
At this stage, loads typical for given geographical location are applied to a numerical model and the structural behaviour is examined. If stress and displacement fields of the structure are in accebtable level, the material strengths are chosen and dimensions of individual elements are calculated. If the structural response is unaccepltable, some adjustments are necessary. More in 3.1.

Detailing
Detailing can have significant impact on the global behaviour of the whole structure and therefore have to be realized precisely according to assumptions done in analysis stage. In some cases a situation may arise, that dimensions of supporting structure shifts the system lines of material far from their original position. When this is true, new form finding and analysis is necessary for the cutting pattern to be correct.

Cutting pattern generation
The production plan of cable-membrane structure is called cutting pattern. Usually, the form finding of tensile structures results in non-developable surface. Therefore the structure can not be projected onto a plane explicitely, but have to be cut into several pieces, flattened and compensated. Fabrication according to these plans will ensure, that the final assembled structure will have the desired shape and prestress. See section 2.4.

In the following sections of chapter 2, the form finding, structural analysis and cutting pattern generation stages will be described from numerical standpoint in detail.

2.1 Form Finding

The most critical factor in the design of tensile structures is the definition of their initial configuration (geometry). This configuration should ensure, that the desired stress field (prestress) is in equilibrium and allows optimal (smooth) force flow within the structure. Several methods have been developed for the initial form finding process, which can be divided into two groups:

- methods based on geometric concept
- methods based on mechanical concept

This work deals with both concepts and uses the geometric concept as an intial procedure for finding the shape of membrane and cables and after that, applies the mechanical concept, which is in classical finite element enviroment able to include stiff elements (trusses, beam or shells) and smoothens the force flow within the whole structure.

During the history, many form finding methods have been developed, especially Force Density method by Scheck [19], which is always a good choice when one wants to find a reliable, clear and predictible method for form finding. This method is easy to implement in a custom software tools and offers nonlinear extensions as well as natural extension for membrane elements named as Natural Force Density method (NFDM) proposed by Pauletti and Pimenta [20]. Barnes developed the dynamic relaxation method [21], which can be used also for form finding of bending active structures or shells. General approach to form finding process using geometrically nonlinear analysis was presented by Argyris, Angelopolous and Bichat [9].

In this work a method based on finite element model using geometric stiffness matrices for both membrane and cable elements is presented.

2.1.1 Geometric form finding method

This method is based on paper by Haber and Abel [11] and is described also in several papers by Baranger [12], [13]. Geometric stiffness method is based on a geometrically nonlinear finite element formulation and on the fact, that the initial equilibrium problem is independent from material. Haber developed the geometric stiffness method in 1982 using a triangular membrane finite element, as will be presented in the following text. It is important to note here, that Force Density method is a special case of the geometric stiffness method and both methods are implemented in much the same way. The designer begins by specifying the topology of the structure in terms of the element connectivity, location of all supported nodes and the

loads acting on the structure. The loads must be shape-independent to preserve the linearity of the solution process. In case of membrane structures formed by prestress only, the loads acting on the structure are assumed to be zero. Solution of the initial equilibrium problem then involves the determination of surface geometry and internal stress distribution that satisfies equilibrium.

For three node triangular element used in this work, the element geometry can be described by its nodal coordinates in global coordinate system as:

$$\boldsymbol{X} = \begin{bmatrix} X_1 & Y_1 & Z_1 & X_2 & Y_2 & Z_2 & X_3 & Y_3 & Z_3 \end{bmatrix}^T \tag{1}$$

Using the procedure described in section 2.2.1, these nine nodal values in global coordinate system can be reduced to six nodal values in local coordinate system as follows:

$$\boldsymbol{x} = \begin{bmatrix} x_1 & y_1 & x_2 & y_2 & x_3 & y_3 \end{bmatrix}^T \tag{2}$$

Element shape functions, that are used to interpolate nodal values of any physical entity into any point of an element surface can be written as:

$$\begin{aligned} N_1 &= \left[\left(x_2 y_3 - x_3 y_2 \right) + \left(y_2 - y_3 \right) x + \left(x_3 - x_2 \right) y \right] / 2A \\ N_2 &= \left[\left(x_3 y_1 - x_1 y_3 \right) + \left(y_3 - y_1 \right) x + \left(x_1 - x_3 \right) y \right] / 2A \\ N_3 &= \left[\left(x_1 y_2 - x_2 y_1 \right) + \left(y_1 - y_2 \right) x + \left(x_2 - x_1 \right) y \right] / 2A \end{aligned} \tag{3}$$

where *A* represents the element surface area, and can be calculated as:

$$A = \frac{1}{2} \left(x_2 y_3 + x_3 y_1 + x_1 y_2 - x_2 y_1 - x_1 y_3 - x_3 y_2 \right) \tag{4}$$

The partial derivatives of the shape functions with respect to cartesian coordinates are:

$$\begin{aligned} \frac{\partial N_1}{\partial x} &= \frac{y_2 - y_3}{2A} = a / 2A \quad & \frac{\partial N_1}{\partial y} &= \frac{x_3 - x_2}{2A} = b / 2A \\ \frac{\partial N_2}{\partial x} &= \frac{y_3 - y_1}{2A} = c / 2A \quad & \frac{\partial N_2}{\partial y} &= \frac{x_1 - x_3}{2A} = d / 2A \end{aligned} \tag{5}$$

$$\frac{\partial N_3}{\partial x} = \frac{y_1 - y_2}{2A} = e/2A \qquad \frac{\partial N_3}{\partial y} = \frac{x_2 - x_1}{2A} = f/2A$$

Where the local coordinate differences *a* through *f* are shown on Fig. 2:

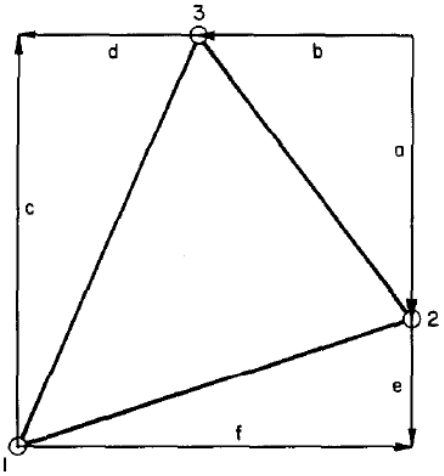


Fig. 2 - Three node constant strain element in local coordinate system

For construction of nodal force vector and element stiffness matrix the following strain-displacement matrices are essential, linear one:

$$\boldsymbol{B}_L = \frac{1}{2A}\begin{bmatrix} a & 0 & c & 0 & e & 0 \\ 0 & b & 0 & d & 0 & f \\ b & a & d & c & f & e \end{bmatrix} \qquad (6)$$

and nonlinear one :

$$\boldsymbol{B}_{NL} = \frac{1}{2A}\begin{bmatrix} a & 0 & 0 & c & 0 & 0 & e & 0 & 0 \\ b & 0 & 0 & d & 0 & 0 & f & 0 & 0 \\ 0 & a & 0 & 0 & c & 0 & 0 & e & 0 \\ 0 & b & 0 & 0 & d & 0 & 0 & f & 0 \\ 0 & 0 & a & 0 & 0 & c & 0 & 0 & e \\ 0 & 0 & b & 0 & 0 & d & 0 & 0 & f \end{bmatrix} \tag{7}$$

The appropriate forms for the vector and matrix of initial prestress at form finding stage are as follows:

$$\hat{\boldsymbol{\sigma}}_0 = \begin{Bmatrix} \sigma_x & \sigma_y & \tau_{xy} \end{Bmatrix}^T \tag{8}$$

and

$$\boldsymbol{\sigma}_0 = \begin{bmatrix} \sigma_x & \tau_{xy} & 0 & 0 & 0 & 0 \\ \tau_{xy} & \sigma_y & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_x & \tau_{xy} & 0 & 0 \\ 0 & 0 & \tau_{xy} & \sigma_y & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_x & \tau_{xy} \\ 0 & 0 & 0 & 0 & \tau_{xy} & \sigma_y \end{bmatrix} \tag{9}$$

Using these matrices the designer can built a set of nonlinear equilibrium equations resulting from the principle of virutual work. Generally the system have the following form:

$$\boldsymbol{K}_G \boldsymbol{x} = \boldsymbol{F} \tag{10}$$

where $\boldsymbol{K_G}$ is a geometric stiffness matrix, which for three node triangular element can be written as:

$$\boldsymbol{K}_G = tA\boldsymbol{B}_{NL}^T \boldsymbol{\sigma}_0 \boldsymbol{B}_{NL} \tag{11}$$

and for cable element:

$$\boldsymbol{K}_G = \frac{N}{L}\begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \end{bmatrix} \tag{12}$$

where t, A, N, L are membrane thickness, membrane element surface area, cable force and cable length respecivelly. $\boldsymbol{x}$ in the Eq. (10) represents a set of nodal coordinates of structure in equilibrium and not a nodal displacements like is usual in standard formulation of finite element method. The expresions for geometric stiffness matrices for both membrane and cable elements don´t have to be transformed to global coordinate system, since the only thing that is neccessary is the element connectivity and stress state. Resulting surface in equilibrium is therefore independent of material properties and intial configuration of the system.
For structures composed entirely of pin-jointed bar elements, the resulting global equations are dentical to the corresponding equations in the Force Density method. This suggests, that presented geometric stiffness method is a generalized version of the Force Density method.

For the case of form finding with prestress only (without external loadings), the load vector $\boldsymbol{F}$ in right side of Eq. (10) is assumed to be zero. System of equations with zero right side don´t have a unique solution, when one tries to solve it by direct method. An iterative conjugate gradient method with preconditioning should be used, or the Eq. (10) should have been rewritten in the terms of static condensation, where the original equation itself have to be splitted into equations belonging to free and fixed nodes:

$$\begin{bmatrix} \boldsymbol{K}_{11} & \boldsymbol{K}_{12} \\ \boldsymbol{K}_{21} & \boldsymbol{K}_{22} \end{bmatrix} \begin{Bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{F}_1 \\ \boldsymbol{F}_2 \end{Bmatrix} \tag{13}$$

or in natural form:

$$\begin{aligned} \boldsymbol{K}_{11}\boldsymbol{x}_1 + \boldsymbol{K}_{12}\boldsymbol{x}_2 &= \boldsymbol{F}_1 \\ \boldsymbol{K}_{21}\boldsymbol{x}_1 + \boldsymbol{K}_{22}\boldsymbol{x}_2 &= \boldsymbol{F}_2 \end{aligned} \tag{14}$$

solving this system for $\boldsymbol{x_2}$:

$$\boldsymbol{x}_2 = \boldsymbol{K}_{22}^{-1}\left(\boldsymbol{F}_2 - \boldsymbol{K}_{21}\boldsymbol{x}_1\right) \tag{15}$$

where $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ collects the nodal coordinates of fixed and free nodes respectivelly, $\boldsymbol{F_1}$ and $\boldsymbol{F_2}$ are the vectors of external loads that belong to fixed and free nodes respectivelly and in the form finding process are assumed to be zero. And finally the $\boldsymbol{K_{ij}}$ (with *i, j* = 1, 2) is a stiffness matrix partitioned into parts belonging to constrained (1) and free (2) degrees of freedom respectivelly.

Although the presented system of equations is very similar to classical formulation of finite element method, other than fom finding elements (cable and membranes) can not be included to it. The reason is that after solution of the system, one will obtain the nodal coordinates, rather than displacements and rotational degrees of freedom are an incompatibile quantitity.

Surface resulting from presented form finding methodology is ready for further design steps such as structural analysis or patterning or can be used as an initial equilibrium surface for mechanical form finding, which has the capability for further smoothening the force flow and taking into account secondary rigid structure (trusses, beams, plates, shells).

2.1.2 Mechanical form finding method

From the mechanical point of view, the form finding process can be seen as a large displacement finite element analysis with form-driving loads. In case of prestressed tensile fabric structures form-driving loads are calculated from prestress only, in case of pneumatic structures the form-driving load is internal pressure and in case of shell structures, the form-driving loads are arbitrary, but mostly the self weight and predominant static loads (dead loads). The resulting displaced configuration of structure represents the structure in force equilibrium. On such a displaced structure, the internal forces are in equilibrium with external loads. When no external loads are present (as in the case of membrane structures), then the internal forces are in equilibrium with each other. During the iterative large displacement finite element analysis the internal forces are calculated from elastic strains and elastic strains are calculated from nodal displacements (see the following section). Since the geometrically nonlinear behaviour of tensile structures indicates, that at least some of the nodes will undergo significant displacements, large strains are possible. Large internal forces resulting from large strains can crash the engineering model (unbalanced forces are

unacceptabilly large) or can give an unrealistic picture of stress state in the membrane or forces in cables. Therefore internal forces resulting from large displacements have to be eliminated and this can be done by setting the Young´s modulus of form-finding elements (membranes and cables) to a fraction of their actual value (for example $E = 10^{-4} – 10^{-5}$, which is $10^{6} – 10^{7}$ times smaller than the actual value for the membrane [17]). It should be noted, that the Young´s modulus of form finding elements is not zero. If a zero modulus is used, the resulting mesh may become highly distorted and will not be usable in load analysis. Furthermore if the inplane stiffness of membrane vanishes, the governing equations may not be solveable. Providing an appropriate intial mesh (generated by procedure described in the previous section) we can start a finte element simulation. After convergence within prescribed precision, the resulting nodal displacements are added to coordinates of initial mesh and the resulting mesh is used in load analysis. In this finite element enviroment trusses, beams and shells can be included in the form finding process in the usual manner, what at the end results in a displaced geometry, which respects the natural force flow.

2.2 Analysis

From mechanical point of view, all loads applied to tensile membrane structures have to be transfered to the secondary (steel structures) or to foundations by pure tensile forces. Since membranes have no physical out-of-plane stiffness, this is possible only by combination of initial prestress, curvature and large displacements reforming the structure under given loads. The mechanical behaviour of the fabric and cables have a nonlinear nature, which can be classified into two categories:

- nonlinear due to large displacements (rotation and translation resulting in geometric nonlinearity)
- nonlinear due to the fact, that neither fabric, nor the cables can withstand compressive forces (stresses)

To solve the mechanical behaviour of such a structure a Total Lagrangian formulation was adopted. This formulation is characterized by the definition of strain and stress fields relative to the initial configuration of the structure at the start of calculation process. The use of fixed reference configuration has the advantage of constant integration domain. At each iteration, the displacement, strain and stress fields are actualized although their definition remains the same. This configuration uses the Green-Lagrange strain tensor for strains and 2nd Piola Kirchhoff tensor for stresses. These tensors are defined with respect to initial configutration before deformation. At the end of the iteration process, the 2nd Piola Kirchhoff stresses have to be transformed to true Cauchy stresses, that can be used for dimensioning.

2.2.1 Membrane element

Simple three node triangular membrane finite element with constat strain property (constant strain triangle) is adopted. This element is the most simple surface finite element and its stiffness matrices can be calculated using one point Gauss integration, or directly, using the simplified procedure, as presented in this work.

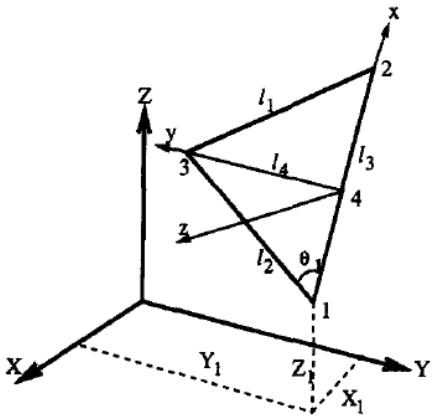


Fig. 3 - Triangular membrane element in 3D space

Fig. 3 shows a typical triangular element in the global coordinate frame (X, Y, Z). An element Cartesian coordinate system (x, y, z) is used to derive the element matrices. Since the element is planar, the z-coordinate is not used in the evaluation of stiffness matrix. It is important to note here, that local coordinate frame is not given only by setting z = 0. Such a simplification would lead to improper descritption of element geometry and at the end would lead to wrong analysis. Local coordinate frame can be obtained using the following procedure:

$$\boldsymbol{A}_{2D} = \boldsymbol{T}_{2D}^{3D} \boldsymbol{A}_{3D} \tag{16}$$

where global coordinate matrix $\boldsymbol{A}_{3D}$:

$$\boldsymbol{A}_{3D} = \begin{bmatrix} X_1 & X_2 & X_3 \\ Y_1 & Y_2 & Y_3 \\ Z_1 & Z_2 & Z_3 \end{bmatrix} \tag{17}$$

coordinate transformation matrix $\boldsymbol{T}_{2D}^{3D}$:

$$\boldsymbol{T}_{2D}^{3D} = \begin{bmatrix} \cos(x,X) & \cos(x,Y) & \cos(x,Z) \\ \cos(y,X) & \cos(y,Y) & \cos(y,Z) \end{bmatrix} \tag{18}$$

It is essential to incorporate all three coordinate directions into transformation matrix, because such a matrix will be used quiet frequently in the further analysis to convert the physical entities between the coordinate

frames. Full transformation matrix for an element with three nodes and three degrees of freedom per node is:

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{\lambda} & 0 & 0 \\ 0 & \boldsymbol{\lambda} & 0 \\ 0 & 0 & \boldsymbol{\lambda} \end{bmatrix} \tag{19}$$

where:

$$\boldsymbol{\lambda} = \begin{bmatrix} \cos(x,X) & \cos(x,Y) & \cos(x,Z) \\ \cos(y,X) & \cos(y,Y) & \cos(y,Z) \\ \cos(z,X) & \cos(z,Y) & \cos(z,Z) \end{bmatrix} \tag{20}$$

With $\cos(y,Z)$ beeing a direction cosine between local y and global Z coordinate axes. Detailed description about seting local coordinate frame can be found in [5] or derived from Fig. 3. Finally the nodal coordinates in local coordinate system are:

$$\boldsymbol{A}_{2D} = \begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{bmatrix} \tag{21}$$

Displacement field of a typical element in local coordinate system is described by nine nodal values:

$$\boldsymbol{u} = \begin{Bmatrix} u_1 & v_1 & w_1 & u_2 & v_2 & w_2 & u_3 & v_3 & w_3 \end{Bmatrix}^T \tag{22}$$

Membrane structures derive their stiffness from the curvature and stress state. These are the only two ways, how the membrane surface with negligeble thickness and thus negligleble flexural (out-of-plane) stiffness can withstand loads, that are acting perpendicular to their surface. Displacements of membrane surface are significant and therefore the small deflection theory of linear elasticity is unaplicable to describe the state of strain and stress correctly. Nonlinear elasticity with quadratic strain-displacement relations must be taken into account.

They can be expressed as:

$$\varepsilon_x = \frac{\partial u}{\partial x} + \frac{1}{2}\left[\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial w}{\partial x}\right)^2\right]$$
$$\varepsilon_y = \frac{\partial v}{\partial y} + \frac{1}{2}\left[\left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 + \left(\frac{\partial w}{\partial y}\right)^2\right] \tag{23}$$
$$\tau_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} + \frac{1}{2}\left[\frac{\partial u}{\partial x}\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\frac{\partial v}{\partial y} + \frac{\partial w}{\partial x}\frac{\partial w}{\partial y}\right]$$

Collected to an element strain vector:

$$\boldsymbol{\varepsilon} = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \tau_{xy} \end{Bmatrix} \tag{24}$$

These strains can be separated into linear and quadratic terms in the following way:

$$\boldsymbol{\varepsilon} = \boldsymbol{B}_0\boldsymbol{u} + \frac{1}{2}\boldsymbol{A}\boldsymbol{\theta} \tag{25}$$

where the linear terms are described by:

$$\boldsymbol{B}_0 = \begin{bmatrix} b_1 & 0 & 0 & b_2 & 0 & 0 & b_3 & 0 & 0 \\ 0 & c_1 & 0 & 0 & c_2 & 0 & 0 & c_3 & 0 \\ c_1 & b_1 & 0 & c_2 & b_2 & 0 & c_3 & b_3 & 0 \end{bmatrix} \tag{26}$$

With b_i and c_i (i = 1, 2, 3) beeing global derivatives of shape functions with following meaning:

$$b_1 = (y_2 - y_3)/2\Delta \quad c_1 = (x_3 - x_2)/2\Delta \qquad 2\Delta = \det\begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix} \tag{27}$$
$$b_2 = (y_3 - y_1)/2\Delta \quad c_2 = (x_1 - x_3)/2\Delta$$
$$b_3 = (y_1 - y_2)/2\Delta \quad c_3 = (x_2 - x_1)/2\Delta$$

and finally the nonlinear terms:

$$\boldsymbol{A} = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial v}{\partial x} & \frac{\partial w}{\partial x} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\partial u}{\partial y} & \frac{\partial v}{\partial y} & \frac{\partial w}{\partial y} \\ \frac{\partial u}{\partial y} & \frac{\partial v}{\partial y} & \frac{\partial w}{\partial y} & \frac{\partial u}{\partial x} & \frac{\partial v}{\partial x} & \frac{\partial w}{\partial x} \end{bmatrix} \tag{28}$$

and

$$\boldsymbol{\theta} = \left\{ \frac{\partial u}{\partial x} \quad \frac{\partial v}{\partial x} \quad \frac{\partial w}{\partial x} \quad \frac{\partial u}{\partial y} \quad \frac{\partial v}{\partial y} \quad \frac{\partial w}{\partial y} \right\}^T \tag{29}$$

thus

$$\delta\boldsymbol{\varepsilon} = \left(\boldsymbol{B}_0 + \boldsymbol{A}\boldsymbol{G}\right)\delta\boldsymbol{u} \tag{30}$$

where

$$\boldsymbol{G} = \begin{bmatrix} b_1 & 0 & 0 & b_2 & 0 & 0 & b_3 & 0 & 0 \\ 0 & b_1 & 0 & 0 & b_2 & 0 & 0 & b_3 & 0 \\ 0 & 0 & b_1 & 0 & 0 & b_2 & 0 & 0 & b_3 \\ c_1 & 0 & 0 & c_2 & 0 & 0 & c_3 & 0 & 0 \\ 0 & c_1 & 0 & 0 & c_2 & 0 & 0 & c_3 & 0 \\ 0 & 0 & c_1 & 0 & 0 & c_2 & 0 & 0 & c_3 \end{bmatrix} \tag{31}$$

Because we are considering large displacements, but small strains, the constitutive relations for linear elastic orthotropic material can be used. Thus we write:

$$\boldsymbol{\sigma} = \boldsymbol{D}\boldsymbol{\varepsilon} + \boldsymbol{\sigma}_0 \tag{32}$$

Where $\boldsymbol{\sigma_0}$ denotes initial stress vector and $\boldsymbol{D}$ is the constitutive matrix:

$$\boldsymbol{D} = \begin{bmatrix} \frac{E_1}{1-\nu_1\nu_2} & \frac{E_1\nu_2}{1-\nu_1\nu_2} & 0 \\ \frac{E_2\nu_1}{1-\nu_1\nu_2} & \frac{E_2}{1-\nu_1\nu_2} & 0 \\ 0 & 0 & G_{12} \end{bmatrix} \tag{33}$$

Constitutive stress-strain matrix as given by Eq. (32) is connected to the material coordinates. Since the element stiffness matrix is written in local coordinate frame, we have to transform the constitutive equations from material coordinates (*w*,*f*) to local coordinates (*x*,*y*). This can be done using following procedure:

$$\bar{\boldsymbol{D}} = \boldsymbol{T}_\theta \boldsymbol{D} \boldsymbol{T}_\theta^T \tag{34}$$

Where $\boldsymbol{T_\theta}$ is the matrix transforming the principal coordinate system of material to the local coordinate system:

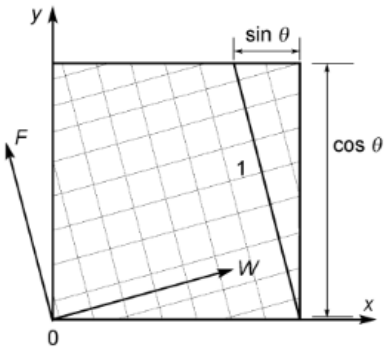


Fig. 4 - Local coordinates (x, y) and material coordinates (w, f)

$$\boldsymbol{T}_\theta = \begin{bmatrix} \cos^2\theta & \sin^2\theta & -2\sin\theta\cos\theta \\ \sin^2\theta & \cos^2\theta & 2\sin\theta\cos\theta \\ \sin\theta\cos\theta & -\sin\theta\cos\theta & \cos^2\theta - \sin^2\theta \end{bmatrix} \tag{35}$$

where θ is the angle between the principal axis of the material (*w*) and the local coordinate (*x*), as shown in Fig. 4 (with clockwise direction being positive). E_1 and E_2 are the elastic modulus in the warp (*w*) and weft (*f*)

directions; G_{12} is the shear modulus; v_1 and v_2 are the Poisson's ratios for the orthotropic materials.
The equilibrium equations for a single element in local coordinate system may now be obtained via the principle of virtual work, as follows:

$$\int_{V^e} \delta\boldsymbol{\varepsilon}^T \boldsymbol{\sigma} dV - \delta\boldsymbol{u}^T \boldsymbol{p} = \boldsymbol{0} \tag{36}$$

Where V^e denotes the element volume and $\boldsymbol{p}$ is the external nodal force vector in the local coordinate system. Substituting Eqs. (25), (30) and (32) into (36) and eliminating $\delta\boldsymbol{u}^T$, we have:

$$\int_{V^e} \left(\boldsymbol{B}_0 + \boldsymbol{AG}\right)^T \left[\bar{\boldsymbol{D}} \left(\boldsymbol{B}_0 \boldsymbol{u} + \frac{1}{2} \boldsymbol{A}\boldsymbol{\theta} \right) + \boldsymbol{\sigma}_0 \right] dV - \boldsymbol{p} = \boldsymbol{0} \tag{37}$$

The above equation is a system of an equilibrium equations for single element in local coordinate system. These equations have to be transformed to global coordinate system and assebmled to global system of equilibrium equations. The assembly process is a typical finite element procedure and can be studied in one of the many finite element lecture notes [36]. Since the global equations will be solved iteratively by the Newton-Raphson method, we will linearize them and assume

$$\boldsymbol{\phi}^i = \int_{V^e} \left(\boldsymbol{B}_0 + \boldsymbol{AG}\right)^T \left[\bar{\boldsymbol{D}} \left(\boldsymbol{B}_0 \boldsymbol{u} + \frac{1}{2} \boldsymbol{A}\boldsymbol{\theta} \right) + \boldsymbol{\sigma}_0 \right] dV - \boldsymbol{p} \tag{38}$$

as the residual term after the *i*-th iteration. For the next step we will use the Newton-Raphson procedure

$$\boldsymbol{\phi}^{i+1} = \boldsymbol{\phi}^i + \frac{\partial \boldsymbol{\phi}^i}{\partial \boldsymbol{u}} \Delta \boldsymbol{u}^i = \boldsymbol{0} \tag{39}$$

From this equation the incremental displacement $\Delta\boldsymbol{u}^i$ can be computed, but first, the partial derivative $\partial\boldsymbol{\phi}^i / \partial\boldsymbol{u}$ have to be computed, which is nothing else, but element tangent stiffness matrix. When rewriting the partial derivative $\partial\boldsymbol{\phi}^i / \partial\boldsymbol{u}$, one will obtain:

$$
\begin{aligned}
&\boldsymbol{k}_T^i = \frac{\partial \boldsymbol{\phi}^i}{\partial \boldsymbol{u}} = \\
&= \int_{V^e} \left(\boldsymbol{B}_0 + \boldsymbol{A}^i\boldsymbol{G}\right)^T \frac{\partial}{\partial \boldsymbol{u}} \left[\bar{\boldsymbol{D}} \left(\boldsymbol{B}_0 \boldsymbol{u}^i + \frac{1}{2}\boldsymbol{A}^i\boldsymbol{\theta}^i \right) \right] dV + \\
&+ \int_{V^e} \frac{\partial}{\partial \boldsymbol{u}} \left[\left(\boldsymbol{B}_0 + \boldsymbol{A}^i\boldsymbol{G}\right)^T \right] \times \left[\bar{\boldsymbol{D}} \left(\boldsymbol{B}_0 \boldsymbol{u}^i + \frac{1}{2}\boldsymbol{A}^i\boldsymbol{\theta}^i \right) + \boldsymbol{\sigma}_0 \right] dV = \\
&= \boldsymbol{k}_e^i + \boldsymbol{k}_g^i
\end{aligned}
\tag{40}
$$

Where $\boldsymbol{k}_e^i$ and $\boldsymbol{k}_g^i$ are the elastic stiffness matrix and the geometric stiffness matrix in iteration *i*, respectively

$$
\boldsymbol{k}_e^i = \int_{V^e} \left(\boldsymbol{B}_0 + \boldsymbol{A}\boldsymbol{G}\right)^T \bar{\boldsymbol{D}} \left(\boldsymbol{B}_0 + \boldsymbol{A}\boldsymbol{G}\right) dV \tag{41}
$$

and

$$
\boldsymbol{k}_g^i = \int_{V^e} \boldsymbol{G}^T \frac{\partial \boldsymbol{A}}{\partial \boldsymbol{u}} \boldsymbol{\sigma} dV = \int_{V^e} \boldsymbol{G}^T \boldsymbol{M}\boldsymbol{G} dV \tag{42}
$$

where

$$
\boldsymbol{M} = \begin{bmatrix}
\sigma_x & 0 & 0 & \tau_{xy} & 0 & 0 \\
0 & \sigma_x & 0 & 0 & \tau_{xy} & 0 \\
0 & 0 & \sigma_x & 0 & 0 & \tau_{xy} \\
\tau_{xy} & 0 & 0 & \sigma_y & 0 & 0 \\
0 & \tau_{xy} & 0 & 0 & \sigma_y & 0 \\
0 & 0 & \tau_{xy} & 0 & 0 & \sigma_y
\end{bmatrix} \tag{43}
$$

For the purpose of computing the constant straint triangle stiffness matrices in local coordinate system numericaly, the following equations can be used:

$$
\boldsymbol{k}_e^i = t\Delta \left(\boldsymbol{B}_0 + \boldsymbol{A}\boldsymbol{G}\right)^T \bar{\boldsymbol{D}} \left(\boldsymbol{B}_0 + \boldsymbol{A}\boldsymbol{G}\right) \tag{44}
$$

and

$$
\boldsymbol{k}_g^i = t\Delta \boldsymbol{G}^T \boldsymbol{M}\boldsymbol{G} \tag{45}
$$

where t and Δ represent the membrane thickness and triangle surface area respectively. Having derived the neccessary terms for stiffness matrix, the

calculation of element nodal forces corresponding to the element displacement vector $\boldsymbol{u}$ can be computed easily:

$$\boldsymbol{f}_e^i = t\Delta\left(\boldsymbol{B}_0 + \boldsymbol{AG}\right)^T \boldsymbol{\sigma} \tag{46}$$

Matrices computed using described procedure are still in local coordinate system and before assembling to global system of equations they must be transformed to global coordinate frame using the transformation matrix $\boldsymbol{T}$, Eq. (19):

$$\boldsymbol{K}_T^i = \boldsymbol{T}^T \boldsymbol{k}_e^i \boldsymbol{T} \tag{47}$$

$$\boldsymbol{F}_e^i = \boldsymbol{T}^T \boldsymbol{f}_e^i \tag{48}$$

$$\boldsymbol{U}_e^i = \boldsymbol{T}^T \boldsymbol{u}_e^i \tag{49}$$

After the convergence of Newton-Raphson iterative technique, one will obtain final deformation field in global coordinate system. Since the derivation of structural matrices used herein correspond to the Total Lagrangian framework, strains used are Green-Lagrange strains and stresses are 2nd Piola Kirchhoff stresses. Such stresses are corresponding to the initial undeformed configuration and have only small physical meaning. For dimensioning and other relevant operations with membrane stress field, the designer is interested in true or so called Cauchy stresses, which are connected with deformed configuration of the structure. To obtain true stresses it is important to define deformation gradient first:

$$\boldsymbol{F} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}} = \begin{bmatrix} \frac{\partial x}{\partial X} & \frac{\partial x}{\partial Y} & \frac{\partial x}{\partial Z} \\ \frac{\partial y}{\partial X} & \frac{\partial y}{\partial Y} & \frac{\partial y}{\partial Z} \\ \frac{\partial z}{\partial X} & \frac{\partial z}{\partial Y} & \frac{\partial z}{\partial Z} \end{bmatrix} = \begin{bmatrix} 1+\frac{\partial u}{\partial X} & \frac{\partial u}{\partial Y} & \frac{\partial u}{\partial Z} \\ \frac{\partial v}{\partial X} & 1+\frac{\partial v}{\partial Y} & \frac{\partial v}{\partial Z} \\ \frac{\partial w}{\partial X} & \frac{\partial w}{\partial Y} & 1+\frac{\partial w}{\partial Z} \end{bmatrix} \tag{50}$$

where $\partial \boldsymbol{X}$ represents the geometry of the structure before deformation and $\partial \boldsymbol{x}$ respresents the structure after deformation, what can be symbolically written as:

$$\delta \boldsymbol{x} = \delta \boldsymbol{X} + \delta \boldsymbol{u} \tag{51}$$

Since membrane element, with thickness negligeble comparing to other dimensions, is a 2D solid element in 3D space, all partial derivatives with

respect to Z coordinate $\partial/\partial Z$ are equal to zero and also transformation of $\partial w/\partial$ to local coordinate frame can be performed. Finally, the deformation gradient for membrane element can be written as:

$$\boldsymbol{F} = \begin{bmatrix} 1+\dfrac{\partial u}{\partial X} & \dfrac{\partial v}{\partial X} \\ \dfrac{\partial u}{\partial Y} & 1+\dfrac{\partial v}{\partial Y} \end{bmatrix} \tag{52}$$

True Cauchy stresses can be obtained from 2nd Piola Kirchoff stresses by using the conversion formula:

$$\boldsymbol{C} = \frac{1}{\det \boldsymbol{F}} \boldsymbol{F}\boldsymbol{\sigma}\boldsymbol{F}^{T} \tag{53}$$

or backwards, 2nd Piola Kirchoff stresses from true Cauchy stresses by an inverse formula:

$$\boldsymbol{\sigma} = \left(\det \boldsymbol{F}\right) \boldsymbol{F}^{-1}\boldsymbol{C}\boldsymbol{F}^{-T} \tag{54}$$

2.2.2 Cable element

Lets have a cable element connectiong nodes $i(X_i, Y_i, Z_i)$ and $j(X_j, Y_j, Z_j)$ in the undeformed configuration as shown in Fig. 5.

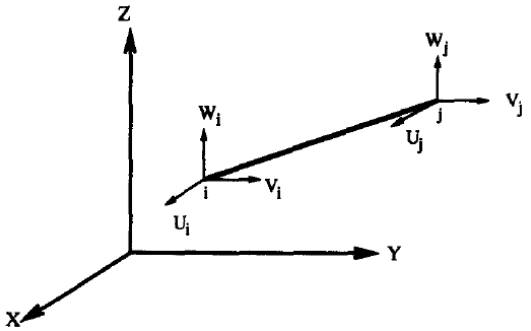


Fig. 5 - A cable element in the global coordinate frame

If nodes *i* and *j* have displacements vectors $[U_i, V_i, W_i]^T$ and $\left[U_j, V_j, W_j\right]^T$ respectively, then we can define the strain along the cable as:

$$\varepsilon = \frac{l_1 - l_0}{l_0} \tag{55}$$

where

$$l_0 = \sqrt{\left[\left(X_j - X_i\right)^2 + \left(Y_j - Y_i\right)^2 + \left(Z_j - Z_i\right)^2\right]} \tag{56}$$

and

$$l_1 = \sqrt{\left[\left(X_j - X_i + U_j - U_i\right)^2 + \left(Y_j - Y_i + V_j - V_i\right)^2 + \left(Z_j - Z_i + W_j - W_i\right)^2\right]} \tag{57}$$

Represent the lengths of cable element before and after deformation. Taking variation of the strain, we have:

$$\delta\varepsilon = \begin{bmatrix} \dfrac{X_j - X_i + U_j - U_i}{l_1}\delta\left(U_j - U_i\right) + \\ \dfrac{Y_j - Y_i + V_j - V_i}{l_1}\delta\left(V_j - V_i\right) + \\ \dfrac{Z_j - Z_i + W_j - W_i}{l_1}\delta\left(W_j - W_i\right) \end{bmatrix} / l_0 \tag{58}$$

What can be written in matrix form as:

$$\delta\varepsilon = \frac{1}{l_0}\boldsymbol{B}\delta\boldsymbol{U} \tag{59}$$

in which

$$\boldsymbol{U} = \begin{bmatrix} U_i & V_i & W_i & U_j & V_j & W_j \end{bmatrix}^T \tag{60}$$

and

$$\boldsymbol{B} = \begin{bmatrix} -C_X & -C_Y & -C_Z & C_X & C_Y & C_Z \end{bmatrix} \tag{61}$$

The direction cosines of the deformed cable C_X, C_Y, C_Z are given by

$$C_X = \frac{X_j - X_i + U_j - U_i}{l_1} \tag{62}$$

$$C_Y = \frac{Y_j - Y_i + V_j - V_i}{l_1} \tag{63}$$

$$C_Z = \frac{Z_j - Z_i + W_j - W_i}{l_1} \tag{64}$$

As for the membrane element, the linear elastic constitutive relation is used for the cable element as:

$$\sigma = E\varepsilon + \sigma_0 \tag{65}$$

where σ_0 is the initial stress in the cable. The constribution of a cable element to the global equilibrium equations can be obtained via the principle of virtual work, as follows:

$$\boldsymbol{\phi}_c = \int_{L^e} \frac{1}{l_0} \boldsymbol{B}^T \sigma A dL = \boldsymbol{B}^T \sigma A \tag{66}$$

Where A is the cross-sectional area of the cable element and L^e denotes the length of the cable element. After linearizing the element equations in preparation for the Newthon-Raphson iteration method we have:

$$\boldsymbol{K}_c = \frac{\partial \boldsymbol{\phi}_c}{\partial \boldsymbol{U}} = A\boldsymbol{B}^T \frac{\partial \sigma}{\partial \boldsymbol{U}} + A\sigma \frac{\partial \boldsymbol{B}}{\partial \boldsymbol{U}} = \frac{EA}{l_0} \boldsymbol{B}^T \boldsymbol{B} + \frac{A\sigma}{l_1} \boldsymbol{C} \tag{67}$$

where

$$\boldsymbol{C} = \begin{bmatrix} C_Y^2 + C_Z^2 & 0 & 0 & -C_Y^2 - C_Z^2 & 0 & 0 \\ 0 & C_X^2 + C_Z^2 & 0 & 0 & -C_X^2 - C_Z^2 & 0 \\ 0 & 0 & C_X^2 + C_Y^2 & 0 & 0 & -C_X^2 - C_Y^2 \\ -C_Y^2 - C_Z^2 & 0 & 0 & C_Y^2 + C_Z^2 & 0 & 0 \\ 0 & -C_X^2 - C_Z^2 & 0 & 0 & C_X^2 + C_Z^2 & 0 \\ 0 & 0 & -C_X^2 - C_Y^2 & 0 & 0 & C_X^2 + C_Y^2 \end{bmatrix} \tag{68}$$

and $\boldsymbol{K_c}$ is an expression for cable tangent stiffness matrix in global coordinate system.

2.2.3 Wrinkling analysis

Because the membrane cannont resist any compressive stresses, wrinkling will occur and stresses in the elements will be redistributed when the external loads give rise to compressive stresses larger then initial tensile

stresses (prestress). In the following lines the principal stress-strain criterion for wrinkling consideration will be described.

First the principal stress and strains at checked node have to be calculated. Principal stresses are:

$$\sigma_1 = \frac{\sigma_x + \sigma_y}{2} + \sqrt{\left(\frac{\sigma_x - \sigma_y}{2}\right)^2 + \tau_{xy}{}^2} = \sigma_{\max} \tag{69}$$

$$\sigma_2 = \frac{\sigma_x + \sigma_y}{2} - \sqrt{\left(\frac{\sigma_x - \sigma_y}{2}\right)^2 + \tau_{xy}{}^2} = \sigma_{\min} \tag{70}$$

and principal strains:

$$\varepsilon_1 = \frac{\varepsilon_x + \varepsilon_y}{2} + \frac{1}{2}\sqrt{\left(\varepsilon_x - \varepsilon_y\right)^2 + \gamma_{xy}{}^2} \tag{71}$$

$$\varepsilon_2 = \frac{\varepsilon_x + \varepsilon_y}{2} - \frac{1}{2}\sqrt{\left(\varepsilon_x - \varepsilon_y\right)^2 + \gamma_{xy}{}^2} \tag{72}$$

Using these values, we are able to make a following check:

$\sigma_1 \le 0$	biaxial wrinkling occurs
$\sigma_2 \le 0$ and $\sigma_1 > 0$	uniaxial wrinkling occurs in second principal direction
$\sigma_2 > 0$	wrinkling does not occur (fully stressed state)

Denoting the angle between local coordinates (x, y) and principal stress coordinates (σ_1, σ_2) as φ and the angle between the material coordinates (w, f) and (σ_1, σ_2) as α, we have the following relationship:

$$\tan\varphi = -\frac{\gamma_{xy}}{\sigma_x - \sigma_2} = -\frac{\gamma_{xy}}{\sigma_1 - \sigma_y} \tag{73}$$

and

$$\alpha = \theta + \varphi \tag{74}$$

graphically:

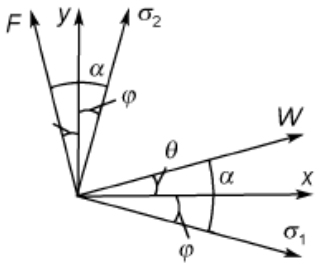


Fig. 6 - Element coordinates systems

According to Eq. (34), the constitutive matrix in the (σ_1, σ_2) coordinate system is formulated as:

$$\tilde{\boldsymbol{D}} = \boldsymbol{T}_\alpha \boldsymbol{D} \boldsymbol{T}_\alpha^{\,T} = \begin{bmatrix} \tilde{D}_{11} & \tilde{D}_{12} & \tilde{D}_{13} \\ \tilde{D}_{21} & \tilde{D}_{22} & \tilde{D}_{23} \\ \tilde{D}_{31} & \tilde{D}_{32} & \tilde{D}_{33} \end{bmatrix} \tag{75}$$

Principal stress vector can be expressed as the product of the constitutive matrix and the strain vector $\boldsymbol{\varepsilon} = \{\varepsilon_1 \quad \varepsilon_2 \quad \varepsilon_{12}\}^T$, where $\varepsilon_{12} \neq 0$ (due to tension-shear coupling effect) in the (σ_1, σ_2) coordinate system:

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ 0 \end{Bmatrix} = \begin{bmatrix} \tilde{D}_{11} & \tilde{D}_{12} & \tilde{D}_{13} \\ \tilde{D}_{21} & \tilde{D}_{22} & \tilde{D}_{23} \\ \tilde{D}_{31} & \tilde{D}_{32} & \tilde{D}_{33} \end{bmatrix} \begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_{12} \end{Bmatrix} \tag{76}$$

This matrix equation can be expanded into 3 simultaneous equations as follows:

$$\sigma_1 = \tilde{D}_{11}\varepsilon_1 + \tilde{D}_{12}\varepsilon_2 + \tilde{D}_{13}\varepsilon_{12} \tag{77}$$

$$\sigma_2 = \tilde{D}_{21}\varepsilon_1 + \tilde{D}_{22}\varepsilon_2 + \tilde{D}_{23}\varepsilon_{12} \tag{78}$$

$$0 = \tilde{D}_{31}\varepsilon_1 + \tilde{D}_{32}\varepsilon_2 + \tilde{D}_{33}\varepsilon_{12} \tag{79}$$

In case of uniaxial wrinkling, when $\sigma_2 \leq 0$ and $\sigma_1 > 0$, the principal stress vector have to be rewritten in the following way:

$$\bar{\boldsymbol{\sigma}} = \{\sigma_1 \quad 0 \quad 0\}^T \tag{80}$$

Note that σ_1 in Eq. (80) is different from that in Eq. (69). The modified principal stress σ_1 and the constitutive matrix can be obtained in the way, described in the next lines.

From Eq. (79) ε_{12} can be written as

$$\varepsilon_{12} = -\frac{\tilde{D}_{31}\varepsilon_1 + \tilde{D}_{32}\varepsilon_2}{\tilde{D}_{33}} \tag{81}$$

Substituting Eq. (81) into Eq. (78) yields

$$\sigma_2 = \left(\tilde{D}_{21} - \frac{\tilde{D}_{23}\tilde{D}_{31}}{\tilde{D}_{33}}\right)\varepsilon_1 + \left(\tilde{D}_{22} - \frac{\tilde{D}_{23}\tilde{D}_{32}}{\tilde{D}_{33}}\right)\varepsilon_2 \tag{82}$$

considering $\sigma_2 = 0$, we have

$$\varepsilon_2 = \frac{\tilde{D}_{21}\tilde{D}_{33} - \tilde{D}_{23}\tilde{D}_{31}}{\tilde{D}_{23}\tilde{D}_{32} - \tilde{D}_{22}\tilde{D}_{33}}\varepsilon_1 \tag{83}$$

According to Eqs. (77), (81) and (83), the modified first principal stress σ_1 is derived as:

$$\sigma_1 = \alpha\varepsilon_1 \tag{84}$$

with

$$\alpha = \frac{1}{\tilde{D}_{23}\tilde{D}_{32} - \tilde{D}_{22}\tilde{D}_{33}} \begin{bmatrix} \tilde{D}_{11}\left(\tilde{D}_{23}\tilde{D}_{32} - \tilde{D}_{22}\tilde{D}_{33}\right) + \\ \tilde{D}_{12}\left(\tilde{D}_{21}\tilde{D}_{33} - \tilde{D}_{23}\tilde{D}_{31}\right) + \\ \tilde{D}_{13}\left(\tilde{D}_{31}\tilde{D}_{22} - \tilde{D}_{21}\tilde{D}_{32}\right) \end{bmatrix} \tag{85}$$

and

$$\varepsilon_1 = \varepsilon_x \cos^2\varphi + \varepsilon_y \sin^2\varphi - \tau_{xy}\cos\varphi\sin\varphi \tag{86}$$

finally, the modified stress vector can be used as described in Eq. (80) and modified constitutive matrix in (σ_1, σ_2) coordinate system can be taken as:

$$\tilde{\boldsymbol{D}}_{\sigma} = \begin{bmatrix} \alpha & 0 & 0 \\ 0 & \eta\alpha & 0 \\ 0 & 0 & \eta\alpha \end{bmatrix} \tag{87}$$

where η is a very small value, and $\eta\alpha$ is usually of order of 10^{-6} in magnitude. A value less than this threshold value would lead to the convergence problem. If the value is greater in magnitude, erroneous solutions may be obtained.

In case of biaxial wrinkling ($\sigma_1 \leq 0$), the principal stress vector is modified to

$$\bar{\boldsymbol{\sigma}} = \begin{Bmatrix} 0 & 0 & 0 \end{Bmatrix}^T \tag{88}$$

and since the slack membrane has no stiffness, the constitutive matrix in (σ_1, σ_2) coordinate system is modified to:

$$\tilde{\boldsymbol{D}}_\sigma = \begin{bmatrix} \eta\alpha & 0 & 0 \\ 0 & \eta\alpha & 0 \\ 0 & 0 & \eta\alpha \end{bmatrix} \tag{89}$$

where $\eta\alpha$ is also set to 10^{-6} in magnitude to meet the requirement of convergence.

The principal stress vector in Eq. (80) or Eq. (88), and constitutive matrix in Eq. (87) or Eq. (89) in the in (σ_1, σ_2) coordinate system should be transformed to local coordinate system as follows:

$$\boldsymbol{\sigma} = \boldsymbol{T}_{-\varphi}\bar{\boldsymbol{\sigma}} \tag{90}$$

$$\bar{\boldsymbol{D}} = \boldsymbol{T}_{-\varphi}\tilde{\boldsymbol{D}}_\sigma \boldsymbol{T}_{-\varphi}{}^T \tag{91}$$

and later on, used in the stiffness matrix and interal force vector construction.

Cable elements are treated in very similar fashion. Axial force in any cable element during any iteration can be computed using well known formula:

$$N = EA\frac{l_1 - l_0}{l_0} + N_0 \tag{92}$$

where N, E, A, l_1, l_0 and N_0 are the axial cable force, Young´s modulus, cross-sectional area, deformed length, undeformed (initial) length and initial pretension respectivelly. When the resulting force is smaller or equal to zero

($N \leq 0$) then the cable force and Young´s modulus are modified in following way:

$$N = 0 \tag{93}$$

and

$$E = 10^{-6} \tag{94}$$

and used for constructing the stiffness matrix and internal force vector for next iteration.

2.3 General solution procedure

General solution procedure for geometrically nonlinear finite element analysis using Total Lagrangian formulation is shown on Fig. 7. This pseudocode can be used for form finding, analysis and, as shown in sections 2.4.1 and 2.4.2, also for geodesic line calculation, flattening and compensating of membrane strips.

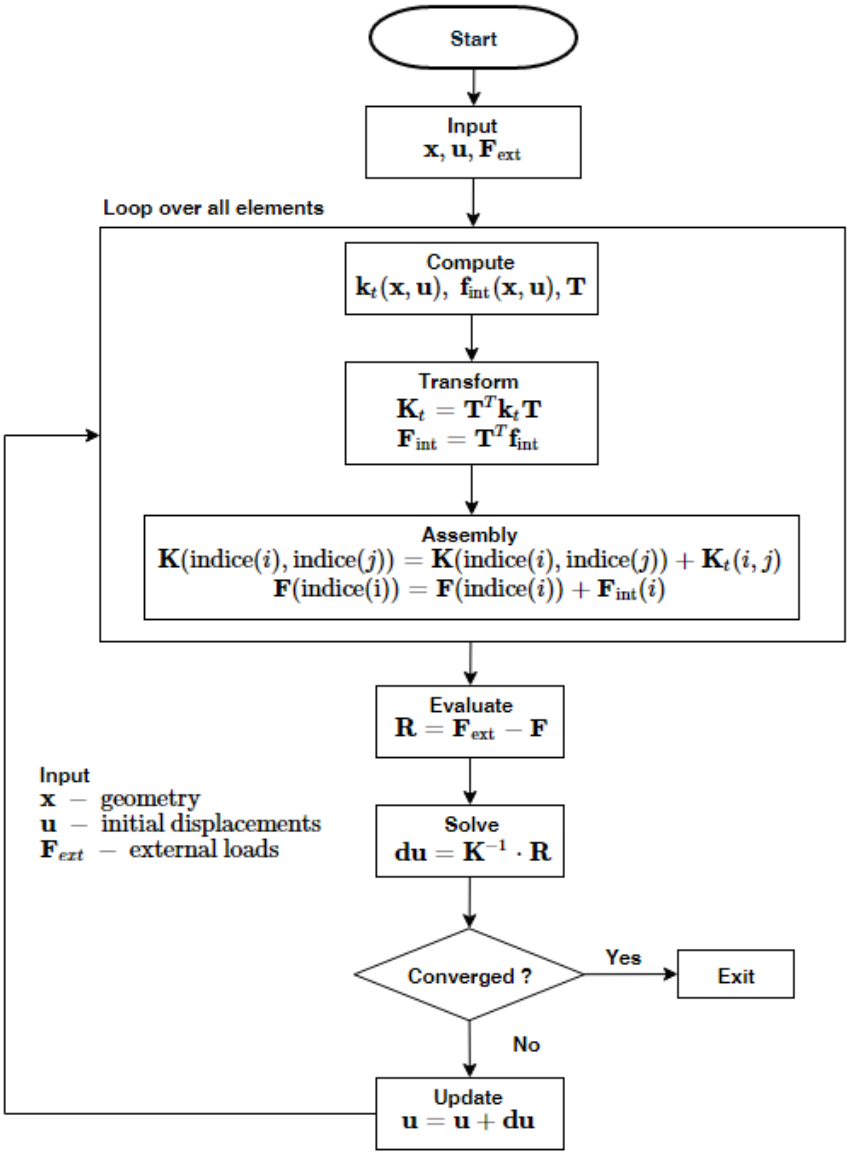


Fig. 7 - General solution procedure

As a convergence test a L^2 norm of displacements increment *du* can be taken.

2.4 Patterning

In general, generating the cutting pattern (which is main production plan) consists of two steps. First, global 3D surface have to be divided into several cloths of a certain width, which respects available fabric roll widths. Secondary, each individual 3D cloth have to be converted plane. This second step requires flattening of 3D doubly curved surface into its 2D image and compensating. Compensating is a process of shrinking planar cloth in a way, that after stretching in construction phase the final 3D surface will be in stress state defined by designer.

This design steps have to be done because most of the membrane structures have larger dimensions then the basic material from which they are made of. They are also doubly curved and prestressed. Curvature of 2D surfaces in 3D space can be described by a single value called Gaussian curvature K. Let´s have an arbitrary surface and assume a single point on it (red dot on Fig. 8). We can draw two perpendicular curves on the surface intersecting in our point. When those curves are drawn in the principal curvature directions (defining directions of principal surface curvature is out of scope of this work and generally it is not needed for the design process), we can measure their radius r_1 and r_2 and calculate their curvatures:

$$R_1 = \frac{1}{r_1} \text{ and } R_2 = \frac{1}{r_2} \tag{95}$$

Gaussian curvature K can be calculated from principal curvatures using following formula:

$$K = R_1 R_2 \tag{96}$$

According to the magnitude of Gaussian curvature we can find out how curved the surface is and according to sign, we can divide surfaces into three basic types:

$K < 0$	negative curvature	anticlastic surfaces
$K = 0$	zero curvature	zeroclastic surfaces
$K > 0$	positive curvature	synclastic surfaces

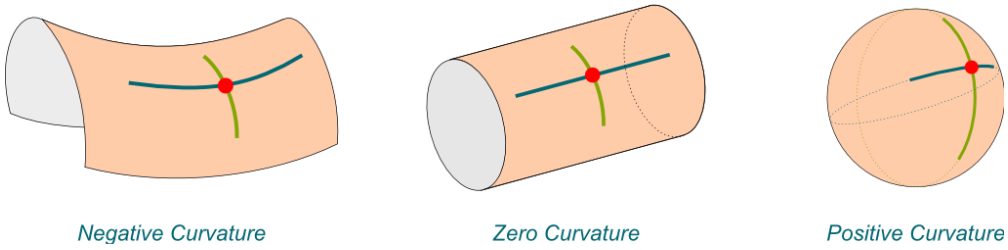


Fig. 8 - Types of surfaces according to their curvature

From the Fig. 8 and Eq.(96) can be easily seen that, if one of the principal curvature is negative (curvatures have oposite sign), resulting surface is anticlastic and therefore have suitable shape for mechanically prestressed membranes. On the other side, if principal curvatures have the same sign, resulting surface is synclastic and can be used for pneumatics and structures prestressed by internal pressure. These surfaces with nonzero curvatures are so called „non-developable“. The term non-developable means, that they can not be simply unfolded to plane, because its internal topology does not allow to do so. Special techniques have to be adopted to flatten these surface to plane (see following sections), but some error will allways be introduced. Finally, if at least one of the principal curvatures is zero (straight line), the resulting surface will be developable (can be unfolded to plane without any error). These surfaces do not fit the requirements that are given to membrane structures.

2.4.1 Cutting lines

To overcome the problem of flattening non-developable surfaces as well as fitting into production limitations (fabric roll widths), the membrane surface have to be divided (cutted) into individual cloths. Generally, cutting can be done in an arbitrary way. Most popular ways of surface cutting are:

- cutting along the intersection of vertical plane-membrane surface
- cutting along the geodesic lines on membrane surface
- arbitrary cutting with respect to architectural requirements

The main disadvantage of cutting along the intersection of vertical plane-membrane surface and arbitrary cutting with respect to architectural requirements is that the flattened cloths have curved edges (or are „banana shaped“), what leads to significant material wastage and the number of seam lines increases.

Fig. 9 - Comparison of non-geodesic (orange) and geodesic (blue) cloth generation

Correctly cut cloths should fulfil several requirements and the cutting lines should be placed with respect to: material properties (copying the warp fabric direction as much as possible, respecting shear stiffness), available material widths and lengths, curvature of the surface (one of the principal curvatures on each cloth should be minimized, so cloth can become nearly developable), main load carrying paths, aesthetical reasons, etc.

This work focuses on cutting the surface along the geodesic lines. Geodesic line between two points on curved 2D surface in 3D space represents their shortest connection and is equivalent to straight line on a plane. There exist more than one geodesic line that connects two same points.

There are many techniques how to generate a geodesic line on a curved surface discretized by flat triangular elements. For example by solving eikonal equation, using floating curve that minimizes its geodesic curvature (GCF principle) or computer aided geometric design (CAGD) principles, that uses 2 stages. At first stage, the initial curve is generated by Dijskstra´s shortest path algorithm or Fast forward marching algorithm. This line is polygonal and respects the edges of finite element mesh. Second stage iteratively improves the initial curve until there is no shorter curve possible. This technique is described in [41].

Technique described in this work is based on mechanical analogy [36]. Prestressed cables are included in finite element mesh along an initial guess for the desired geodesic lines and mechanical method of form finding (described in section 2.1.2) is performed. The spatial movement of prestressed cables during form finding is constrained in a way, that only tangential displacements are allowed and displacements normal (perpendicular) to the surface are ignored. During the mechanical form finding process the geometry proceeds to a configuration with a minimum potential energy. Two main strategies can be adopted:

- geodesic line calculation during form finding
- geodesic line calculation on found form

If the designer wants to find the geodesic line during the form finding process, the initial guess of its final position have to be made. For the computational purposes, it is neccessary that geodesic lines have to be created from egdes of finite element mesh.

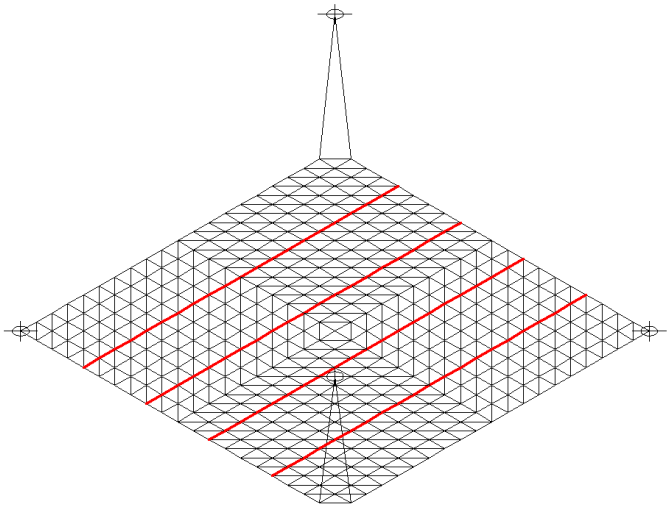
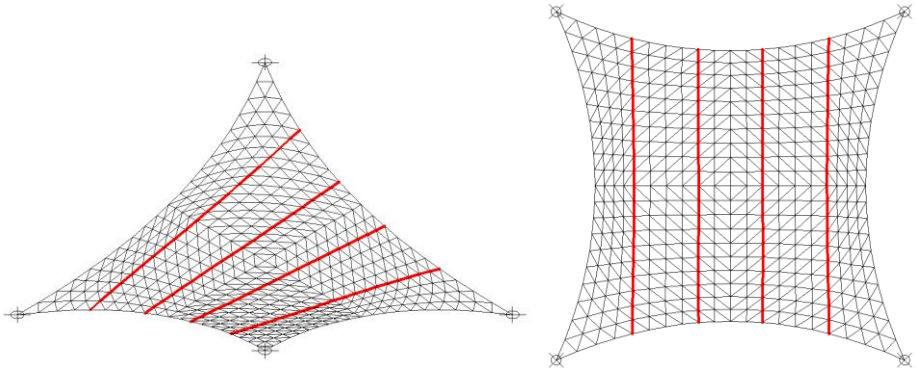
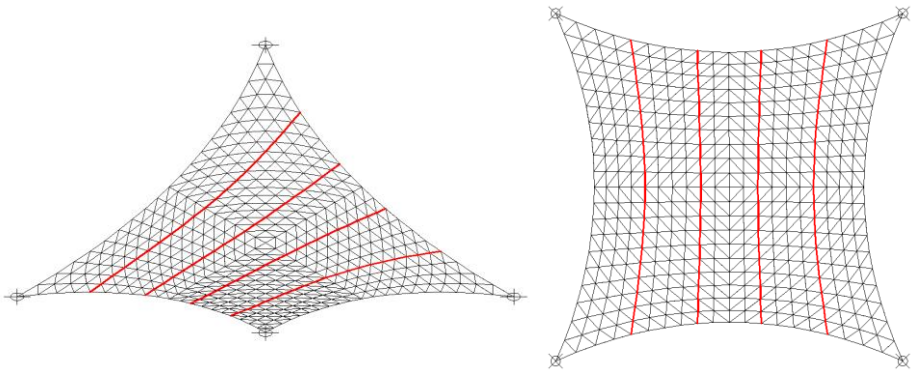


Fig. 10 - Initial guess of geodesic lines

The difference between finite element mesh topology on surface form found with and without geodesic lines can be seen from the following Fig. 11 a)-d).



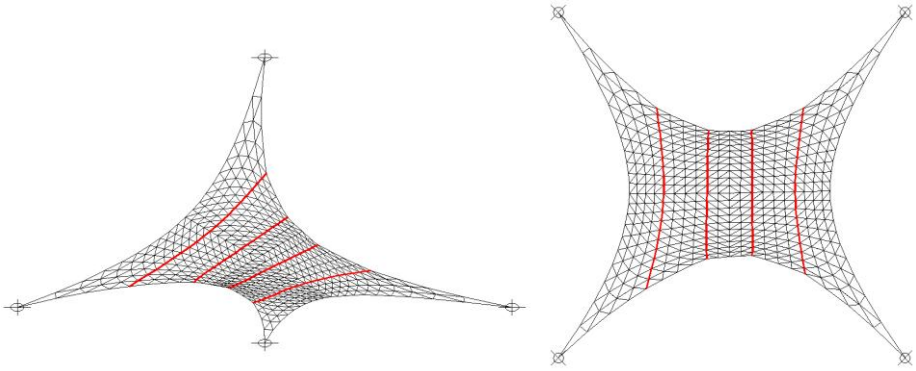
a) b)



c) d)

Fig. 11 - Form found surface with /a) and b)/ and without /c) and d)/ geodesic lines

The in-plane movement of geodesic line nodes was achieved by neglecting the force acting perpendicular to surface. Geodesic lines are represented by high tensioned cables with zero mass and stiffness. The pretension in this case was set to ratio 1:100 (membrane : geodesic line). It is important to note here, that the nodal forces from geodesic pretension are set to zero at its starting and end point. In case this is not done, the pretension will change the form in an unacceptable way.



a) b)

Fig. 12 - Mesh deformation due to tensioning of endpoints of geodesic lines

When the geodesic line have to calculated on a found form and designer would like to use the mechanical approach, two possible strategies are available. First technique is based on a contact formulation between prestressed cable and membrane mesh. This technique is quiet sophisicated and is out of scope of this work. Second strategy is very similar to technique, when geodesic line is calculated during form finding. At this second approach we consider all the nodes belonging to stiff elements (tie cables, trusses, beams, shells) as fixed, also all nodes that belong to edge, or ridge and valley cables are restrained. Only the nodes belonging to membrane

surface are allowed to move. The nonlinear finite element analysis with high tensioned geodesic lines is performed. As well as in the previous technique, nodes on geodesic lines are allowed to move only in direction tangential to the surface (normal movements are ignored).

When the initial position of cutting lines are not known in advance, and therefore final geodesic line can´t be calculated during form finding, designer can use following approach:

1. surface is form found in usual way
2. designer will chose start and end point of the final cutting line
3. shortest path respecitng the edges of the mesh is calculated
4. calculated shortest path is assumed as an inital position of geodesic line and nonlinear finite element analysis is performed, where only the nodes belonging to membrane are allowed to move

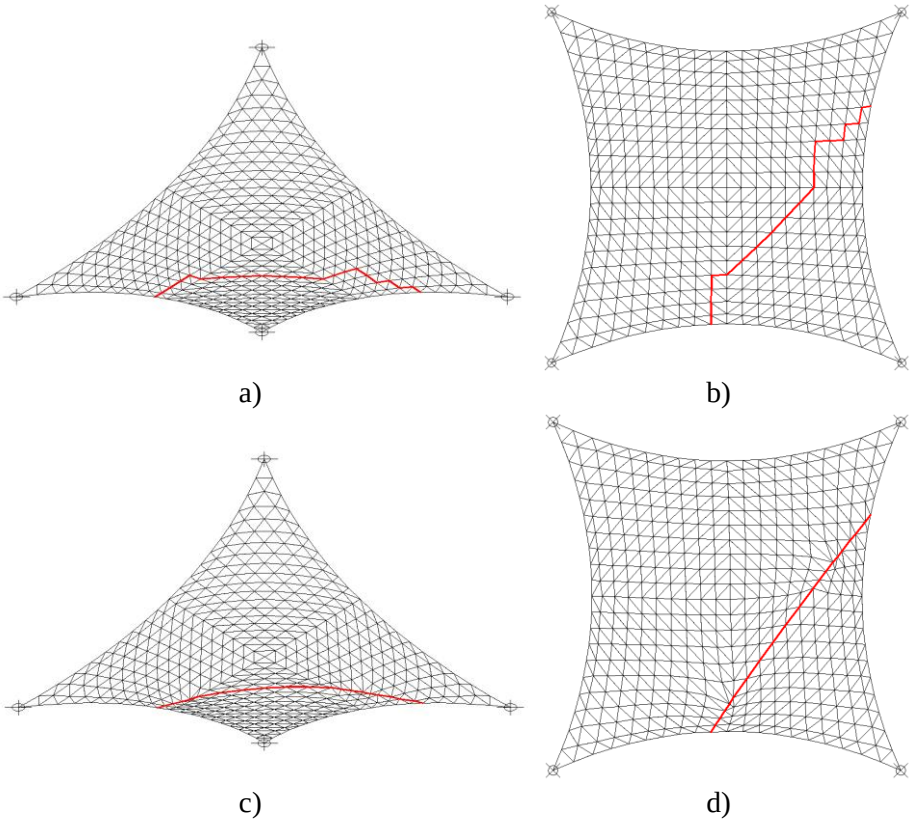


Fig. 13 - Initial shortest path /a) and b)/ and final geodesic line /c) and d)/ - hypar geometry

Shortest path between starting and end point of cutting line consisting of mesh egdes can be calculated in a following way:

1. starting point A and end point B are selected
2. find all the nodes N connected to node A with the mesh edge
3. for every node N calculate the euclidean distance to node B
4. node P with shortest distance to node B is added to the initial shortest path
5. set P = A and continue until A = B

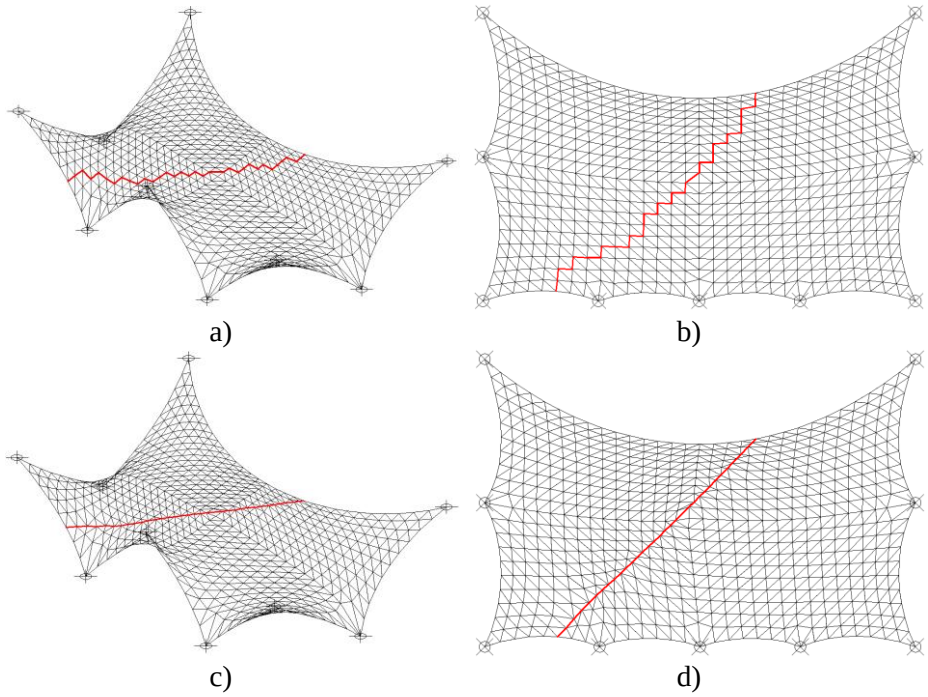


Fig. 14 - Initial shortest path /a) and b)/ and final geodesic line /c) and d)/ - general geometry

As can be seen from Fig. 13 and Fig. 14, this technique works sufficiently good even for initial shortest paths that are far from geodesic lines.

Nodal displacements, that are tangential to the surface can achieved by setting all the nodal forces belonging to prestressed geodesic lines perpendicular to the surface to zero. Nodal forces belonging to prestressed geodesic lines in global coordinate system can be calculated from Eq. (97):

$$\boldsymbol{F}_i = \boldsymbol{B}^T N \tag{97}$$

where $\boldsymbol{F_i}$ is 6x1 vector of nodal forces from actual geodesic line force N. This force is equal to the initial prestress of geodesic line. First three rows of $\boldsymbol{F_i}$ represent the X, Y and Z components of force N in global coordinate system belonging to the first node. Last three rows represent the same, but for the second node. These three components of force N have to be transformed to the local coordinate system of the surface at respective node. This local coordinate system can be calculated as an average of the local coordinate systems of the adjacent triagles:

$$\tilde{\boldsymbol{T}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{\lambda}_i \tag{98}$$

where $\tilde{\boldsymbol{T}}$ is local coordinate system at a surface node, n is a number of adjacent triangles, $\boldsymbol{\lambda}_i$ is the matrix of direction cosines for triangle i (Eq. (20)). Nodal forces in local surface coordinate system are then calculated as:

$$\begin{Bmatrix} F_{Lx} \\ F_{Ly} \\ F_{Lz} \end{Bmatrix} = \tilde{\boldsymbol{T}} \begin{Bmatrix} F_{ix} \\ F_{iy} \\ F_{iz} \end{Bmatrix} \tag{99}$$

where F_{Lx}, F_{Ly} and F_{Lz} are the geodesic line nodal forces in local surface coordinate system. F_{Lz} is set equal to zero, and the resulting vector is then transformed back to global coordinate system:

$$\begin{Bmatrix} F_{ix} \\ F_{iy} \\ F_{iz} \end{Bmatrix} = \tilde{\boldsymbol{T}}^T \begin{Bmatrix} F_{Lx} \\ F_{Ly} \\ 0 \end{Bmatrix} \tag{100}$$

Resulting force components F_{ix}, F_{iy} and F_{iz} are then assembled to the global force vector $\boldsymbol{F}$ in the usuall manner and next iteration of finite element analysis is performed. After the solution of finite element system, the designer will obtain a new increment of displacements $\boldsymbol{du}$. Very high prestress of geodesic lines and its initial path, which is far from equilibrium can cause high values of $\boldsymbol{du}$. These high values can cause overlapping of finite elemnt mesh and can „run out of the surface“ even if only tangential displacements are allowed. Therefore an introduction of damping is essential. Good results were obtained when the maximum value of displacement increment was set to $\alpha\, L_{min}$. Where L_{min} is length of the

shortest mesh edge and α is a user defined constant from interval $\langle 0,1 \rangle$. The value of α depends on mesh quality and initial path topology.

Sufficiently good results were obtained when the technique of tangential-movement-only was applied only to nodes belonging to geodesic lines. If the designer wants to perform precise analysis, this procedure can be applied for every node of the mesh. In such a case a computational time increases very rapidly.

Techniques presented in this chapter are based on mechanical analogy and they work with edges of finite element mesh. This is an advantage when compared to the geodesic line calculation using the CAGD approchaes, which uses the edge intersections or set of nodes independent of finite element mesh. After finding a geodesic line over the surface with CAGD techniques, the remeshing is neccessary, what finally results in an iterative approach or losing the computational data connected to the finite element mesh. When using the mechanical approach, no remeshing is neccessary and designer can use the same mesh for further design stages.

2.4.2 Flattening

As already mentioned, doubly curved anticlastic surfaces are non-developable. This means, that they can´t be unfolded into plane without an error in surface area. Therefore, the procedure called flattening have to be used. Flattening generally means the minimization of difference between 2D planar cloth and its 3D pattern. The amount of potential difference between 2D and 3D surface area during flattening increases with increasing double curvature. Proper cutting of the surface can, among the others, minimize one of the surface curvatures, what can minimize potential error.

During the history, many techniques have been used to flatten 3D cloths to 2D plane. These techniques were based on physical models, models that are minimizing least squares differences between 2D and 3D geometry, mass-spring models, or geometrical models which preserve intrinsic characteristics of triangulation. In the present day following three procedures are used in the most of the cases:

- simple triangulization
- optimization techniques
- mechanical approach

Simple triangulization method is based on an idea, that a general non-developable surface can become developable after remeshing each membrane strip with linear triangles.

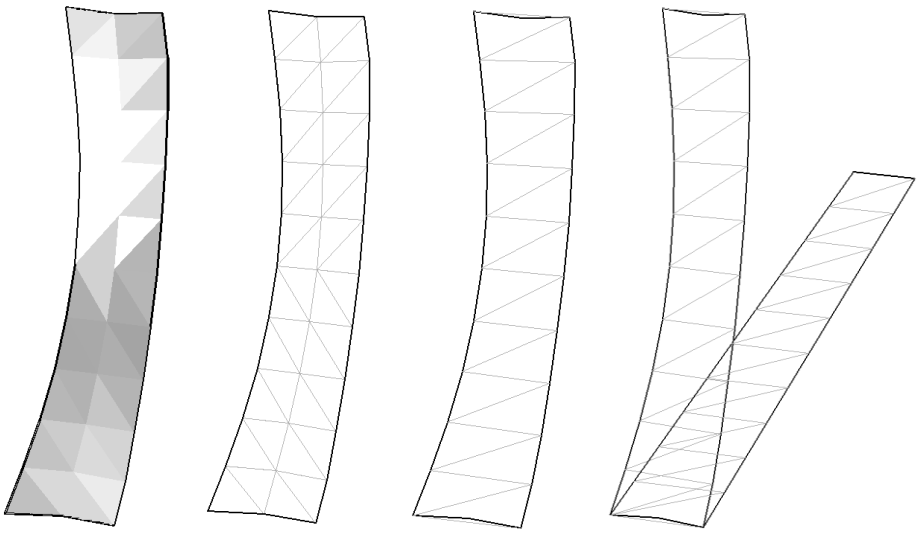


Fig. 15 - Simple triangulization flattening procedure

The basic procedure of simple triangulization flattening (unfolding) can be seen from Fig. 15. Initial 3D surface shown on Fig. 15 a) is cut out of the structure. When the strip is cut from the structure by the procedures described in previous chapter, the finite element mesh (shown on Fig.13 b)) describing the surface is the same, as was used in form finding and analysis. The strip is then remeshed with simple triangles (Fig. 15 c)) which are creating developable surface, which can be simply unfolded (Fig. 15 d)), or redrawn in a plane without an error in surface area. This technique is purely geometric and therefore fast. The main disadvantage lies in remeshing step. After remeshing of intially curved surface with flat triangular elements, all the informations about curvature between cutting lines is lost. Also the edges have to be treated separatelly, since they can carry the information about the curvature of edge cables or cut-offs neccessary for detailing. Hence, this technique have to be used carefully.

Optimization techniques offer many strategies, how to flatten doubly curved spatial membrane strips. They are based on minimization of the square difference between 2D and 3D pattern. Nodal coordinates of flattened pattern are taken as optimization variables in all cases. Objective function, that have to be minimized, can be chosen by the designer freely. Most popular objective functions are:

- minimization of the difference between mesh edge lengths

$$P = \sum_{i=1}^{n_{edges}} \left(L_{3D,i} - L_{2D,i} \right)^2 \tag{101}$$

- minimization of difference in surface area

$$P = \sum_{i=1}^{n_{triangles}} \left(A_{3D,i} - A_{2D,i} \right)^2 \tag{102}$$

- minimization of stress diference

$$P = \sum_{i=1}^{n_{triangles}} \left(\sigma_{3D,i} - \sigma_{2D,i} \right)^2 \tag{103}$$

And many others can be used. For sufficiently precise flattening, the unconstrained optimization techniques (described in section 3.5) can be used. If the designer wants to have more control over the flattening proces, the constrained minimization (described in section 3.6) is an essential choice. In case of constrained minimization, all potential objective functions can be used as constraints and hence the flattening process can be driven in a way, that fits the structural needs. When minimization of difference in lengths or surface area is performed, the optimization task is purely geometrical. In case of minimization of the stress difference between 2D and 3D surfaces, the material properties and their orientation can be taken into account. This is the most suitable way of optimization for anisotropic fabric structures, since it is possible to describe the change in geometry with respect to warp and weft directions. This requires the evaluation of stresses and strains, what makes the definition of objective function more complicated and costly.

After the membrane strips are flattend preciselly enough and their prestress wasn´t considered in the process of flattening, the compensation of flattened patterns have to be performed. Compensation is the process of making the flattened strips smaller in a way, that after their extension (elongation) during the construction phase, they will be in final position and correctly prestressed. Simple geometrical way of compensation can be performed in the following way:

$$\begin{aligned} \boldsymbol{x}_{compensated} &= (1-a)\,\boldsymbol{x}_{original} \\ \boldsymbol{y}_{compensated} &= (1-b)\,\boldsymbol{y}_{original} \end{aligned} \tag{104}$$

where a and b are compensation factors in warp and weft directions respectivelly and $\boldsymbol{x}$ and $\boldsymbol{y}$ are the nodal coordinates before and after compensation (according to their subscript). Their value depends on the material properties and the level of prestress and should be taken from biaxial tests. It is important to notice here, that the x and y cartesian coordiante system have to be oriented in warp and weft directions.

Mechanical approach of surface flattening is based on nonlinear finite element method [36]. Generally, using this approach it is possible to generate flattened patterns only, or flattened and already compensated patterns. The basic idea consist of two stages: press and release. The first phase is based on generating a planar image of a spatial strip. This initial planarization can be done in any way, just the triangulation have to have the same topology (connectivity) and overlapping of element have to be prevented. From the coordinate difference between initially flattened pattern and its 3D parent, the strains, stresses and nodal forces can be calculated (Eqs. (25), (32), (46)). These forces are taken as an external loads, which will drive the relaxation process during the second phase. During this second phase the nonlinear finite element analysis is performed and the final displaced configuration represent the flattened pattern. A clear advantage of this method is that the same mesh as for form finding and analysis is used and therefore no simplifications in geometry is introduced. Patterns generated with mechanical approach are flattened with respect to material properties and their orientation (in case of anisotropic material). This method (like all finite element calculations) requires, that statically determinate supports have to be introduced. If designer sets less or improper supports, the finite element calculation could not converge or even not run. On the other hand, if there are too much supports (more than 3 degrees of freedom in plane are constrained), statical determinity is lost and pattern will be flattened incorrectly (since some nodes, which should change their position, are fixed).

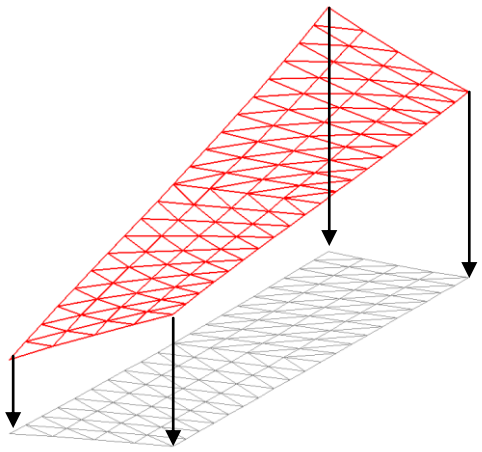


Fig. 16 - Initial flattening of spatial strips

Initial displacements needed for mechanically motivated flattening can be for every single triangle element mathematically written as:

$$\boldsymbol{u} = \begin{Bmatrix} u_1 \\ v_1 \\ w_1 \\ u_2 \\ v_2 \\ w_2 \\ u_3 \\ v_2 \\ w_3 \end{Bmatrix} = \begin{Bmatrix} x_1^{3D} - x_1^{2D} \\ y_1^{3D} - y_1^{2D} \\ z_1^{3D} - z_1^{2D} \\ x_2^{3D} - x_2^{2D} \\ y_2^{3D} - y_2^{2D} \\ z_2^{3D} - z_2^{2D} \\ x_3^{3D} - x_3^{2D} \\ y_3^{3D} - y_3^{2D} \\ z_3^{3D} - z_3^{2D} \end{Bmatrix} \tag{105}$$

Note that these displacements are calculated in global coordinate system and have to be transformed to local coordinate system of element to perform further calculations. Internal forces (integrated on planar configuration) are calculated using these displacements and Eqs. (25), (32), (46). After their assembly to global force vector, is this vector taken as vector of external loads and second phase is performed.

Hypar shown on Fig. 11 c) can be divided with respect to cutting lines to five independent pieces and these pieces are initially flattened according to previous description.

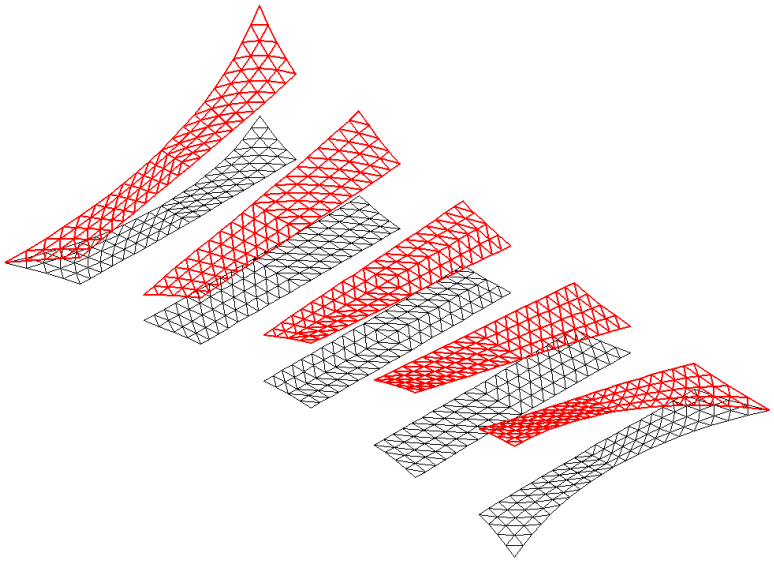


Fig. 17 - Spatial strips and thier planar mapping (initially flattened)

Original 3D hypar had total surface area 292,4656 m^2, intially flattened patterns 281,1460 m^2 (obtained by setting all z-coordinates to zero) and flattened pattern relaxed by nonlinear finite element analysis 292,0648 m^2. The error between 2D and 3D strips is therefore (292,4656 - 292,0648) / 292,4656 * 100 % = 0,137 %. These results were obtained with finite element analysis converegence criteria (based on residual forces) equal to 10^{-5}, which is very precise. When designer wants to obtain even higher precision, the iterative approach can be adopted. After convergence of finite element method, the difference in 2D and 3D geometry is measured again and new load vector for new nonlinear analysis is assembled. This process is repeated until there is only negligible change between these loops. In this particular example, 20 loops were used and the final flattened area was 292,4674 m^2, which means an error (292,4656 - 292,4674) / 292,4656 * 100 % = 0,00061 %.

Until now, only flattening without compensation have been presented. Compensation can be simply performed during this flattening process by setting an initial stresses to planar patterns equal to prestress of 3D strips. Every triangle can have different initial stresses. Therefore triangles will be differently compensated and this compensation will respect the orientation of material directions.

To demonstrate the differences between constant geometric compensation defined by Eq. (104) and stress based compensation described in previous text, the following example have been chosen. Lets have a conical

membrane strip, which is prestressed by 2 kN/m in *x* direction and linearly varying prestress in *y* direction (Fig. 18).

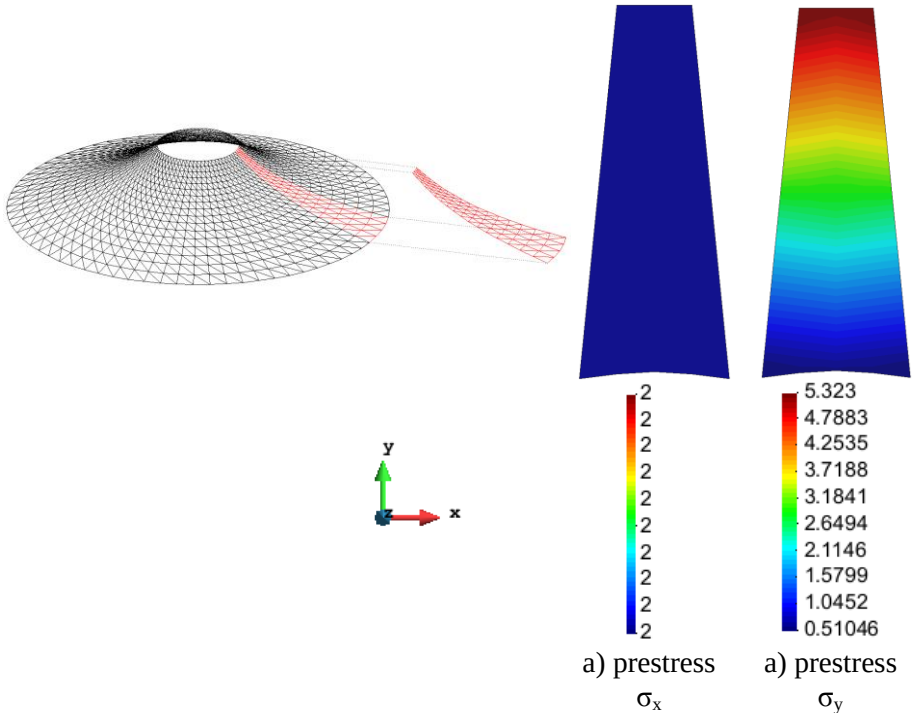


Fig. 18 - Arbitrary conical strip with uneven prestress state

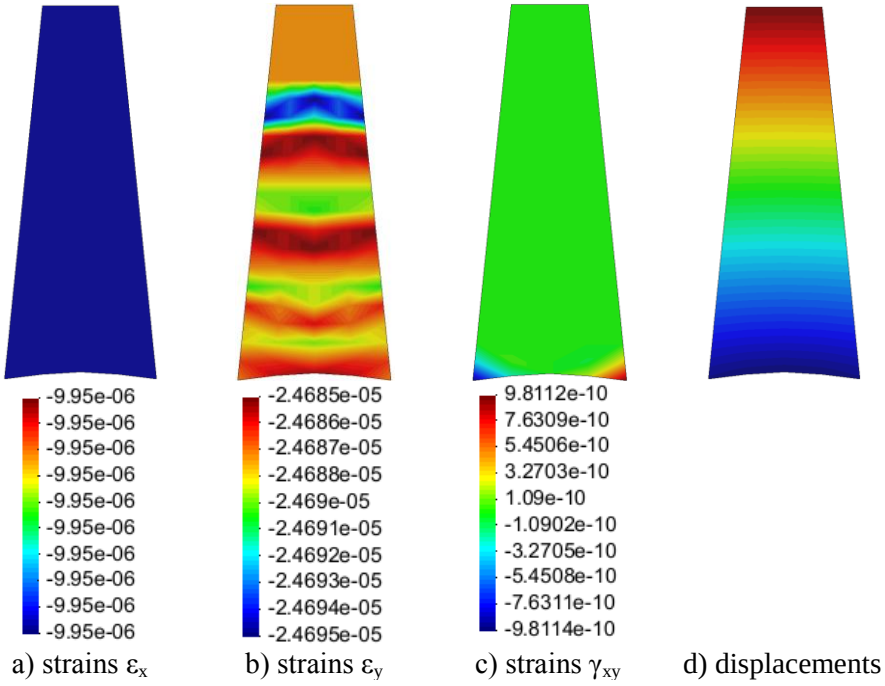


Fig. 19 - Strains resulting from constant compensation

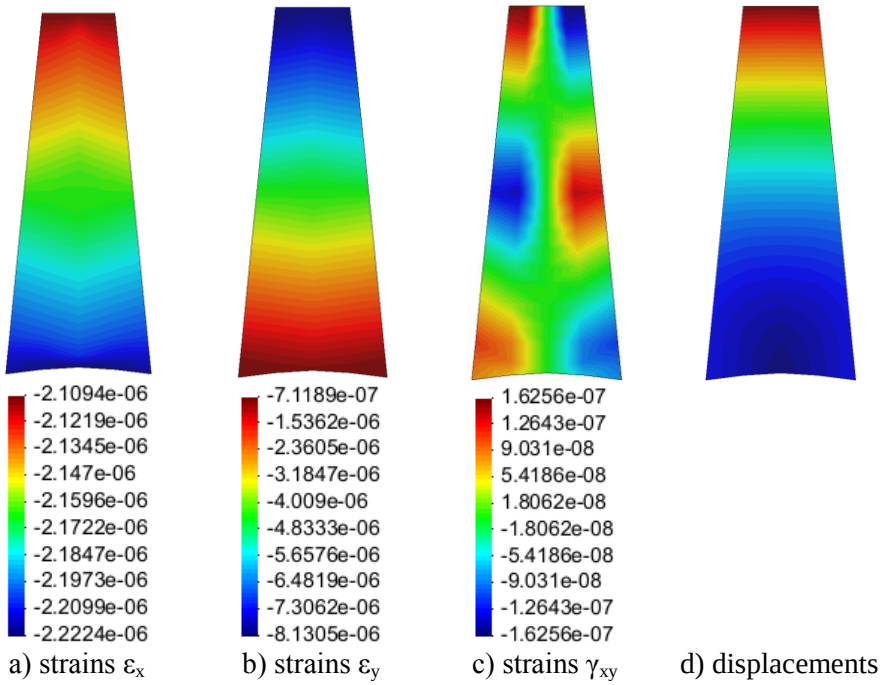
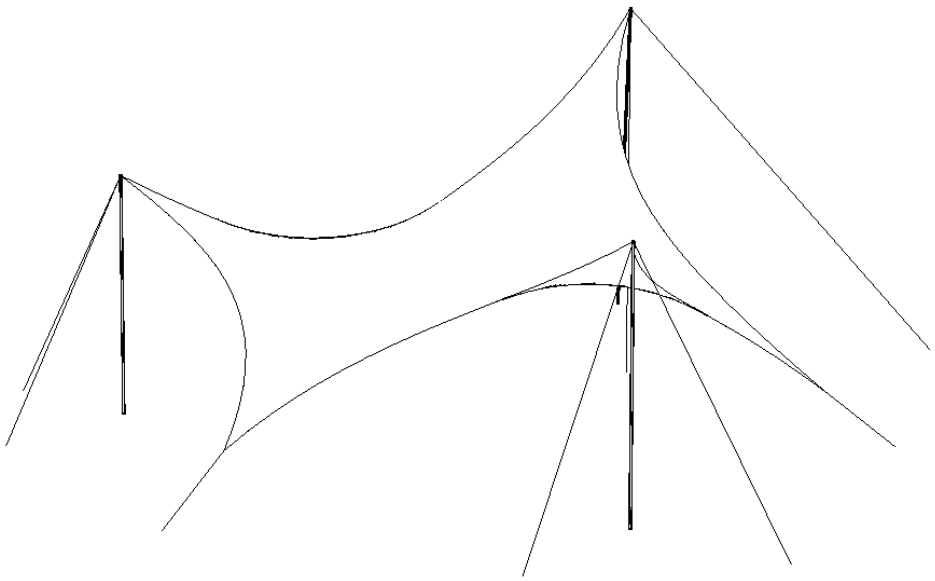


Fig. 20 - Strains resulting from stress based compensation

Fig. 19 presents the strains of a typical membrane strip after the constant geometric compensation. In this particular case the compensation factors 1 % and 2,5 % were used for *x* and *y* directions respectivelly. The final strains in warp and weft (*x* and *y* directions respectivelly) are constant (note that chaotic color fields in case of ε_y are the result of a numerical noise). The shear strains are zero all over the surface, what means that the varying stress state over the membrane strip is not considered and shear stiffness of fabric is not taken into account.

On the other hand, Fig. 20 shows the strain field of fabric after stress based compensation. These results show, that the fabric is compensated more in the areas of higher stress. The difference is about 5 % in warp (constantly stressed) direction and 1142 % in weft (variably stressed) direction. The variation of stress state in weft direction is 1042 %, what means, that also the prestress in *x* direction is taken into account by Poisson constants. Finally, nonzero shear strains results in a different geometry of compensated strips.



3 Structural optimization

Mathematical optimization is a general tool, which is focused on finding the best possible solution of a problem within given design space and under given constraints. Its generality is based on fact, that the solution process is independend of optimization problem. Therefore it is often used in the field of physics, economics and all types of engineering applications like aerospace, automotive, naval or civil engineering. For the proper use of optimization techniques, it is importnat to distinguish the basic terminology:

Design variables: parameters to be changed in order to improve to design (for example nodal coordinates, surface prestress)
Objective function: function of design variables to be optimized (for example minimization of nodal displacements, maximization of structural stiffness)
Inequality constraints: one sided conditions that must be satisfied for the design to be acceptable (for example maximum stress have to be smaller than material strength)
Equality constraints: precise conditions that must be satisfied for the design to be acceptable (very rare in structural optimization problems, for example structural thickness)
Side constraints: bounds on design variables that limit the region of search for the optimum (maximum and minimum values for the design variables)
Feasible design: design that satisfies all constraints
Infeasible design: design that violates one or more constraints
Optimum design: the set of design variables corresponding to minimum (maximum) of the objective function and satisfying all constraints

Structural optimization can be divided into different groups according to different criteria. Based on computational strategy, optimization algorithms can be divided to:

- gradient based optimization
- heuristic optimization

Gradient based optimization algorithms are based on iterative improvement of intial (user defined) design. A search direction is computed as a result of sensitivity analysis and new set of design variables is obtained by changing current design in this direction. On the other hand, heuristic optimization

algorithms require several intial designs. Objective function is evaluated for every design and new set of design variables is obtained as a combination of several best performing designs.
Whether the optimization task is constrained or not, the algorithms can be divided into:

- unconstrained optimization
- constrianed optimization

Last, but not least, depending on the geometric feature, the structural optimization problems can be divided into three classes:

- sizing optimization
- shape optimization
- topology optimization

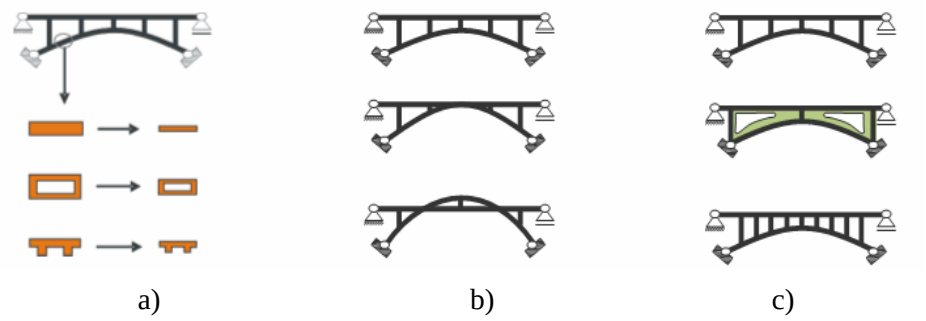


Fig. 21 - Types of structural optimization problems - a) sizing, b) shape, c) topology optimization

Sizing optimization is a class of optimization problems, where the structural thickness or cross-sectional areas are taken as design variables. This type of optimization doesn´t change the shape or topology of a structure, its main goal is to change cross sections in a way, that their utilization is as high as desired. Shape optimization, on the other hand, keeps the cross sections and topology unchanged and changes the geometry to achieve optimization goals. Topology optimization introduces a „density like“ variable, which can have arbitrary value (usually 0 or 1) and controls the topology of the structure. The result of topology optimization is an optimal distribution of material over the given domain. Several other classifications based on different criteria can be found in [28].

This work is focused on gradient based shape optimization, both constrained and unconstrained. Nevertheless, all presented procedures and strategies can be used generally, not only for civil engineering problems.

The capabilities of shape optimization techniques in engineering design can be clearly described by following Fig. 22:

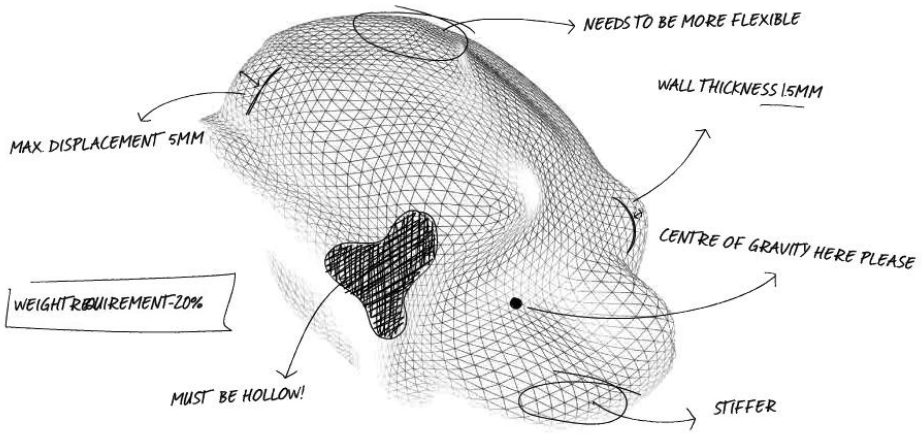


Fig. 22 - General mechanical system

Assume that we have a general structure, which behaviour can be described by computational mechanics. The structural response can be evaluated by finite element method and both primary (displacements, buckling factor) and secondary (stresses, internal forces) unknowns are calculated. According to results, neccessary improvements (for example the displacements have to be smaller) and constraints (maximum stress have to be lower then material strength) can be defined. For shape optimization, nodal coordinates of structure, control points of shape paramterization or shape creating elements (forces, stresses) are taken as design variables.

Typical objective functions used in structural optimization:

- weight or volume
- compliance (structural stiffness)
- stress leveling
- frequency

Typical constraint functions:

- weight or volume
- displacement
- stresses
- frequency

Generally, more then one objective function can be optimized at once. When this is true, the process is then called multi-objective optimization.

3.1 Introduction to optimization of tensile structures

When a tensile fabric structure is designed, the designer first defines an initial shape (form finding step) according to architectural demands. As a second step, mechanical analysis under given load cases is performed and the response of the structure is studied. If the results of mechanical analysis are unacceptable due to aesthetics or safety reasons, the designer should then carry out neccessary design modifications. Since the response of tensile fabric structure is nonlinear and therefore unpredictable, making a correct design modifications is not straightforward. The most typical unacceptable effects of loads are:

- over-stressed areas damaging the fabric
- under-stressed areas causing wrinkling and slack cables (low stiffness)
- large displacements

To avoid these problems some modifications to initial design have to be made. These modifications can be classified in two categories:

- fundamental modifications leading to different architecture or structural behaviour
- correction of design parameters

First category of possible modifications requires a direct intervention of designer and calls for a new creative process. These modifications are very difficult to automate. Second category represents slight modifications of design parameters, such as nodal coordiantes of supports, geometry of secondary structure, material fibres orientation or prestress. These parameters can be taken as a design variables in optimization process and their values, leading to optimal structural behaviour, can be obtained.

Most common objective functions (goals of optimization) regarding to tensile structures are:

- minimize the displacements of certain nodes (high / low points, tripod, too big movement of flat surface)

$$f(\boldsymbol{x}) = \sum_{i=1}^{n_{nodes}} \sqrt{u_i^2 + v_i^2 + w_i^2} \qquad (106)$$

where u_i, v_i and w_i are the displacements of node i in X, Y and Z directions respectivelly

- minimize the stress difference in fabric

$$f(\boldsymbol{x}) = \sum_{i=1}^{n_{elements}} \left(\sigma_i - \sigma_{AVG}\right)^2 \tag{107}$$

where σ_i is stress in *i-th* finite element and σ_{AVG} is an average stress in fabric in measured direction

- minimize the strain energy (maximize stiffness)

$$f(\boldsymbol{x}) = \frac{1}{2}\, \boldsymbol{f}_{ext}{}^T \boldsymbol{u} \tag{108}$$

where $\boldsymbol{f}_{ext}$ is a vector of external forces and $\boldsymbol{u}$ is a vector of corresponding nodal displacements

To obtain acceptable results, the structure have to be constrained. Constraints are imposed mostly on stresses, cable forces or nodal displacements:

- stress constraints

$$\sigma_i \le \sigma_{\max} \qquad \frac{\sigma_i}{\sigma_{\max}} - 1 \le 0 \tag{109}$$

$$\sigma_{\min} \le \sigma_i \quad \text{or} \quad \frac{\sigma_{\min}}{\sigma_i} - 1 \le 0 \tag{110}$$

$$0 \le \sigma_i \qquad -sign\left(\sigma_i\right) \le 0 \tag{111}$$

- cable force constraints

$$N_i \le N_{\max} \qquad \frac{N_i}{N_{\max}} - 1 \le 0 \tag{112}$$

$$N_{\min} \le N_i \quad \text{or} \quad \frac{N_{\min}}{N_i} - 1 \le 0 \tag{113}$$

$$0 \le N_i \qquad -sign\left(N_i\right) \le 0 \tag{114}$$

- nodal displacement constraints

$$|d_i| \le d_{\max} \qquad \text{or} \qquad \frac{|d_i|}{d_{\max}} - 1 \le 0 \tag{115}$$

In case of more then one objective function $f(x)$ have to be optimized (multi-objective optimization), the following formula for weighting sum should be used:

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} w_i f_i(\boldsymbol{x}) \tag{116}$$

where w_i is a weighting factor and n is a number of distinct objective functions.

All necessary informations about the structural optimization task are now ready and such defined system is prepared to perform neccessary calculations. Engineering practice in the beggining of 21st century is able to solve optimization problems with hundreeds-thousands of design variables and milions of constrains. These possibilities greatly exceeds the needs of usual tensile structures.

3.2 Basic concepts of optimization

General mathematical formulation of the optimization problem is most often written as:

$$\min f(\boldsymbol{x}) \tag{117}$$

subject to

$$g_j(\boldsymbol{x}) \le 0 \qquad j = 1 \ldots m \qquad \text{inequality constraints} \tag{118}$$

$$h_k(\boldsymbol{x}) = 0 \qquad k = 1 \ldots l \qquad \text{equality constraints} \tag{119}$$

$$\boldsymbol{x}_i^{\min} \le \boldsymbol{x}_i \le \boldsymbol{x}_i^{\max} \qquad i = 1 \ldots n \qquad \text{side constraints} \tag{120}$$

where $f(\boldsymbol{x})$ is a scalar representation of objective function, $g_j(\boldsymbol{x})$, $h_k(\boldsymbol{x})$ are the vectors of inequality and equality constraints respectivelly and $\boldsymbol{x}_i^{min}$, $\boldsymbol{x}_i^{max}$ are the lower and upper bounds on design variable $\boldsymbol{x}_i$. $\boldsymbol{x}$ itself is a vector of design variables in form:

$$\boldsymbol{x} = \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{Bmatrix} \tag{121}$$

The objective function $f(\boldsymbol{x})$ as well as the constraint functions $c_j(\boldsymbol{x})$ and $h_k(\boldsymbol{x})$ may be linear or nonlinear functions of design variables $\boldsymbol{x}$. These functions may be explicit or implicit in $\boldsymbol{x}$ and may be evaluated by any analytical or numerical techniques available or can be measured experimentally. A convenient form of constraining functions is that of Eqs. (109)-(115). Except for special classes of optimization problems, it is important that these functions be continuous and have continuous first derivatives in $\boldsymbol{x}$.

The above formulation of the optimization problem is not unique and various other formulations are presented in the literature. For example, if we wish to maximize an objective function (in case of buckling factor or eigenfrequencies optimization), we simply minimize the negative of original objective, minimize $-f(\boldsymbol{x})$. Similary, in case of inequality constraints, the designer may wish to have some value of $g_j(\boldsymbol{x})$ greater then zero, realisation is then the same as in case of objective function.

As already mentioned in previous chapters, scope of this work is on gradient based optimization procedures. This type of algorithms is often called a local methods. This menas that finding of global optimum (best of all) is not guaranteed. This will be demonstrated on a following Fig. 23:

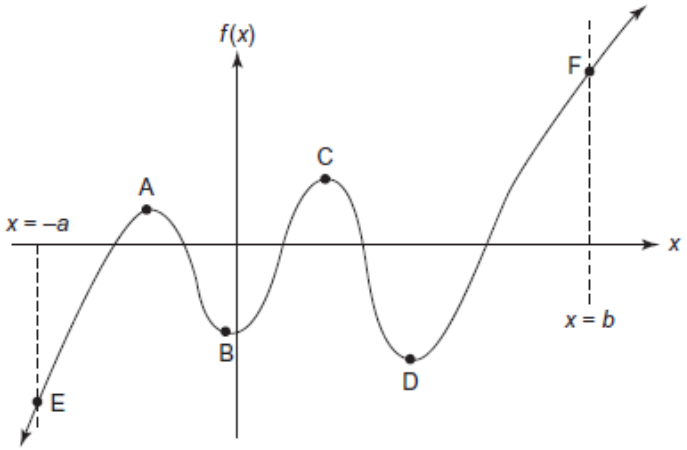


Fig. 23 - Representation of optimum points

Assume, that we have a general function $f(\boldsymbol{x})$ and its lower and upper bounds $-a$ and b respectivelly. Points A and C are so called local maxima, points B and D are local minima, point F is a constrained global maximum and point E is constrained global minimum. The term „constrained“ is importnat, because it tells us, that the function can have a lower (or higer) value, but at least one constraint will be violated and the solution will be therefore infeasible and unacceptable. The term local minimum/maximum means, that there are no feasible points in a close neighborhood with better (lower/higher) objective function values.

Gradient based optimization is an iterative procedure and most of its algorithms require an initial set of design variables $\boldsymbol{x}$ (initial design) to be specified. Beginning from this starting point, the design is updated iteratilvely. The most common updating procedure is given by equation:

$$\boldsymbol{x}^q = \boldsymbol{x}^{q-1} + \alpha \boldsymbol{S}^q \tag{122}$$

where q is the iteration number and $\boldsymbol{S}$ is a vector of search direction in the design space. The scalar quantity α defines the distance that we wish to move in direction $\boldsymbol{S}$ and is also known as step length (see section 3.4). Eq. (122) is very similar to the usual engineering approach, where designer change an existing design a little, to achieve some improvement. The determination of α and $\boldsymbol{S}$ plays a fundamental role in the optimization process. Determination of the search direction $\boldsymbol{S}$ differs according to

algorithm used, or whether the constrained or unconstrained optimization is performed.

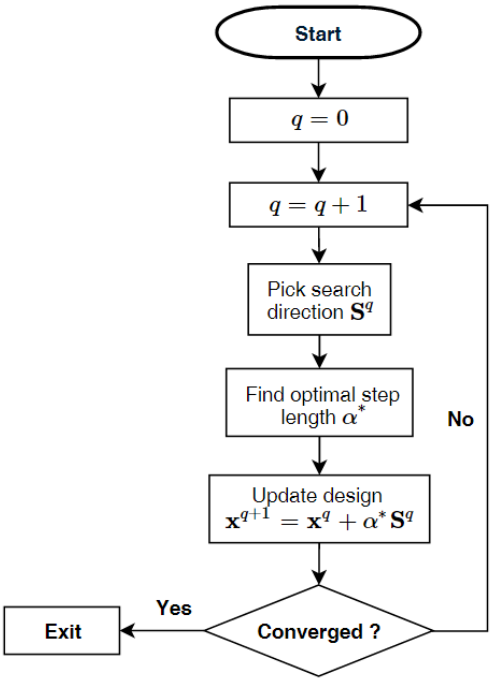


Fig. 24 - General minimization strategy

Fig. 24 provides a general flowchart for multivariable optimization. As seen from the figure, an actual optimization program consists of three major components:

1) determine search direction in which to search (section 3.5 and 3.6)
2) find optimal step length – one dimensional search (section 3.4)
3) convergence check (section 3.5.1)

The reason why gradient based methods converge only to a local minimum (which can, but also don´t have to, be a global minimum) is hidden in Eq. (122) and Fig. 23. When the designer starts the optimization process with initial design corresponding to point A on Fig. 23, two possible directions for decreasing the $f(\boldsymbol{x})$ are possible. In case the movement will be in $+x$ direction, the function will decrease and optimization algorithm will converge to local minimum point B. In case of movement in $-x$ direction, the optimization algorithm will converge to constrained global minimum point E. Both directions can decrease the objective function (which is the goal of optimization), but only one converges to global minimum. Which direction the optimization algorithm decides to go, depends on the sensitivity of the problem to the change of design variables.

3.3 Sensitivity analysis

Procedure to identify the sensitivity of the problem to change of design parameters is called sensitivity analysis. Sensitivities can be understood as a first derivatives of the problem with respect to design variables. When one wants to calculate the derivatives of a function with respect to more then one variable, the results can be organized to a vector, which is called gradient. Mathematically, the gradient is written as:

$$f\left(\boldsymbol{x}\right)=\frac{\partial f\left(\boldsymbol{x}\right)}{\partial \boldsymbol{x}}=\begin{Bmatrix}\frac{\partial f\left(\boldsymbol{x}\right)}{\partial x_1}\\ \frac{\partial f\left(\boldsymbol{x}\right)}{\partial x_2}\\ \frac{\partial f\left(\boldsymbol{x}\right)}{\partial x_3}\\ \vdots\\ \frac{\partial f\left(\boldsymbol{x}\right)}{\partial x_n}\end{Bmatrix} \tag{123}$$

Each component of the gradient vector (vector of first derivatives) can be differentiated again with respect to each variable, to obtain second partial derivatives for the function $f(\boldsymbol{x})$.

$$\nabla^2 f\left(\boldsymbol{x}\right)=\boldsymbol{H}\left(\boldsymbol{x}\right)=\frac{\partial^2 f\left(\boldsymbol{x}\right)}{\partial x_i x_j}=\begin{bmatrix}\frac{\partial^2 f\left(\boldsymbol{x}\right)}{\partial x_1 x_1} & \cdots & \frac{\partial^2 f\left(\boldsymbol{x}\right)}{\partial x_1 x_n}\\ \vdots & \ddots & \vdots\\ \frac{\partial^2 f\left(\boldsymbol{x}\right)}{\partial x_n x_1} & \cdots & \frac{\partial^2 f\left(\boldsymbol{x}\right)}{\partial x_n x_n}\end{bmatrix} \tag{124}$$

with $i=j=1\ \ldots n$. Matrix of second derivatives $\boldsymbol{H}(\boldsymbol{x})$ of a function $f(\boldsymbol{x})$ is known as Hessian matrix and if $f(\boldsymbol{x})$ is twice continously differentiable, then $\boldsymbol{H}(\boldsymbol{x})$ is symmetric.

On several occasions we must diferentiate a vector function of *m* variables with respect to *n* design variables. In constrained optimization problems we often have to diferentiate the vector of constraints $\boldsymbol{g}(\boldsymbol{x})$ with respect to all design variables $\boldsymbol{x}$. Differentiation of vector $\boldsymbol{g}(\boldsymbol{x})$ with respect to one component of vector $\boldsymbol{x}$ results in gradient vector, such as $\nabla g(x_i)$. After

diferentiation with respect to all design variables, resulting gradients can be arranged to a matrix as its columns. This matrix is of size $m \times n$, and is refered to as gradient matrix of $\boldsymbol{g}(\boldsymbol{x})$, or in constrained optimization problems, as a Jacobian matrix of constraints $\boldsymbol{J}\left(\boldsymbol{g}(\boldsymbol{x})\right)$.

$$\begin{aligned} &\nabla g\left(\boldsymbol{x}\right) = \boldsymbol{J}\left(\boldsymbol{g}\left(\boldsymbol{x}\right)\right) = \\ &= \frac{\partial g_i\left(\boldsymbol{x}\right)}{\partial \boldsymbol{x}} = \left[\nabla g_1\left(\boldsymbol{x}\right) \quad \nabla g_2\left(\boldsymbol{x}\right) \quad \dots \quad \nabla g_n\left(\boldsymbol{x}\right)\right] \end{aligned} \tag{125}$$

In mathematics, derivatives of general function $\boldsymbol{f}(\boldsymbol{x})$ with respect to design variable x is most often numerically approximated by finite difference method. Forwad difference approximation of $\partial f_i(\boldsymbol{x})/\partial x_j$ at design $\boldsymbol{x}$ is:

$$\frac{\partial f_i\left(\boldsymbol{x}\right)}{\partial x_j} = \frac{f_i\left(\boldsymbol{x} + e_j \cdot \Delta x\right) - f_i\left(\boldsymbol{x}\right)}{\Delta x} \tag{126}$$

where $e_j = [0 \quad ... \quad 0 \quad 1 \quad 0 \quad ... \quad 0]$ is an identifier of design variable with respect to which is the differentiation performed and Δx is a perturbation. Forward difference approximation of second order derivatives based on function calls can be computed from:

$$\frac{\partial^2 f_i\left(\boldsymbol{x}\right)}{\partial x_i x_j} = \frac{f_i\left(\boldsymbol{x} + e_i \cdot \Delta x_i + e_j \cdot \Delta x_j\right) - f_i\left(\boldsymbol{x} + e_i \cdot \Delta x_i\right) - f_i\left(\boldsymbol{x} + e_j \cdot \Delta x_j\right) - f_i\left(\boldsymbol{x}\right)}{\Delta x_i \Delta x_j} \tag{127}$$

More accurate, but on the other hand, more expensive for calculation time are the central difference approximations. Central difference approximation of $\partial f_i(\boldsymbol{x})/\partial x_j$ at design $\boldsymbol{x}$ is:

$$\frac{\partial f_i\left(\boldsymbol{x}\right)}{\partial x_j} = \frac{f_i\left(\boldsymbol{x} + e_j \cdot \Delta x\right) - f_i\left(\boldsymbol{x} - e_j \cdot \Delta x\right)}{2\Delta x} \tag{128}$$

and for second order derivatives:

$$\frac{\partial^2 f_i(\boldsymbol{x})}{\partial x_i x_j} = \frac{A - B - C - D}{4\Delta x_i \Delta x_j} \tag{129}$$

with

$$A = f_i\left(\boldsymbol{x} + e_i \cdot \Delta x_i + e_j \cdot \Delta x_j\right) \tag{130}$$

$$B = f_i\left(\boldsymbol{x} + e_i \cdot \Delta x_i - e_j \cdot \Delta x_j\right) \tag{131}$$

$$C = f_i\left(\boldsymbol{x} - e_i \cdot \Delta x_i + e_j \cdot \Delta x_j\right) \tag{132}$$

$$D = f_i\left(\boldsymbol{x} - e_i \cdot \Delta x_i - e_j \cdot \Delta x_j\right) \tag{133}$$

Numerical approximation of second order derivatives is very expensive and therefore is not recommended for structural optimization problems. For the optimization methods requiring second order information about the problem, the BFGS approximation of Hessian matrix is much more practical. Although finite difference method of derivative approximations and sensitivity analysis may be very inaccurate and expensive, they at least have one advantage – they are very easy to implement and when finite element analysis code is used as a „black box“, they are the only possible choice.
Generally, there are three groups of methods for sensitivity analysis:

- numerical methods (approximate, for example finite differences)
- analytical methods (exact, but complex)
- hybrid (combination of previous two, for example semi-analytical methods)

Theoretical background of the analytical and hybrid methods is out of scope of this work, but in serious practical applications it is recommended to use them, wherever possible. Interested reader is addressed to [23], [26] and [40].

3.4 Step length search

After the search direction $\boldsymbol{S}^q$ is calculated (see sections 3.5 and 3.6), the length of the movement in this direction have to be determined. This is an unconstrained optimization problem with one design variable, mostly termed as α. Mathematically, this task can be written as:

$$\alpha^* = \arg\min_{\alpha=\langle 0,s\rangle} f\left(\boldsymbol{x} - \alpha \boldsymbol{S}^q\right) \tag{134}$$

where α^* is an optimal length of the step and s is an upper bound given to this length. Maximum value of s depends on the type of the problem, but $s = 1$ is often used. Length of the step have to provide a sufficient decrease of objective function $f(\boldsymbol{x})$ and its calculation have to be as cheap as possible.

Eq. (134) can be solved using zero-order methods, i.e. methods, that don´t require derivatives. Famous zero-order methods are bisection (interval-reducing method) and golden section method, which algorithms are provided on Fig. 25 and Fig. 26 respectivelly. These methods are quite expensive, because they require many function evaluations and therefore are suitable for smaller problems, or problems with low-cost objective function evaluation. They also can be used to set initial function values for more effective step length search methods, such as polynomial approximations.

There are also different kind of algorithms to determine step length known as inexact line search methods. These methods don´t guarantee the maximum decrease of objective function, but offer sufficient decrease in computationally effective manner. Several inexact line search procedures have been developed, from which the most famous are:

- Armijo´s rule
- Wolfe conditions
- Goldstein test

Exact line searches (as given in Fig. 25 and Fig. 26) can be time consuming, and therefore inexact procedures are recomended. The underlying theory can be studied at [38] or [39].

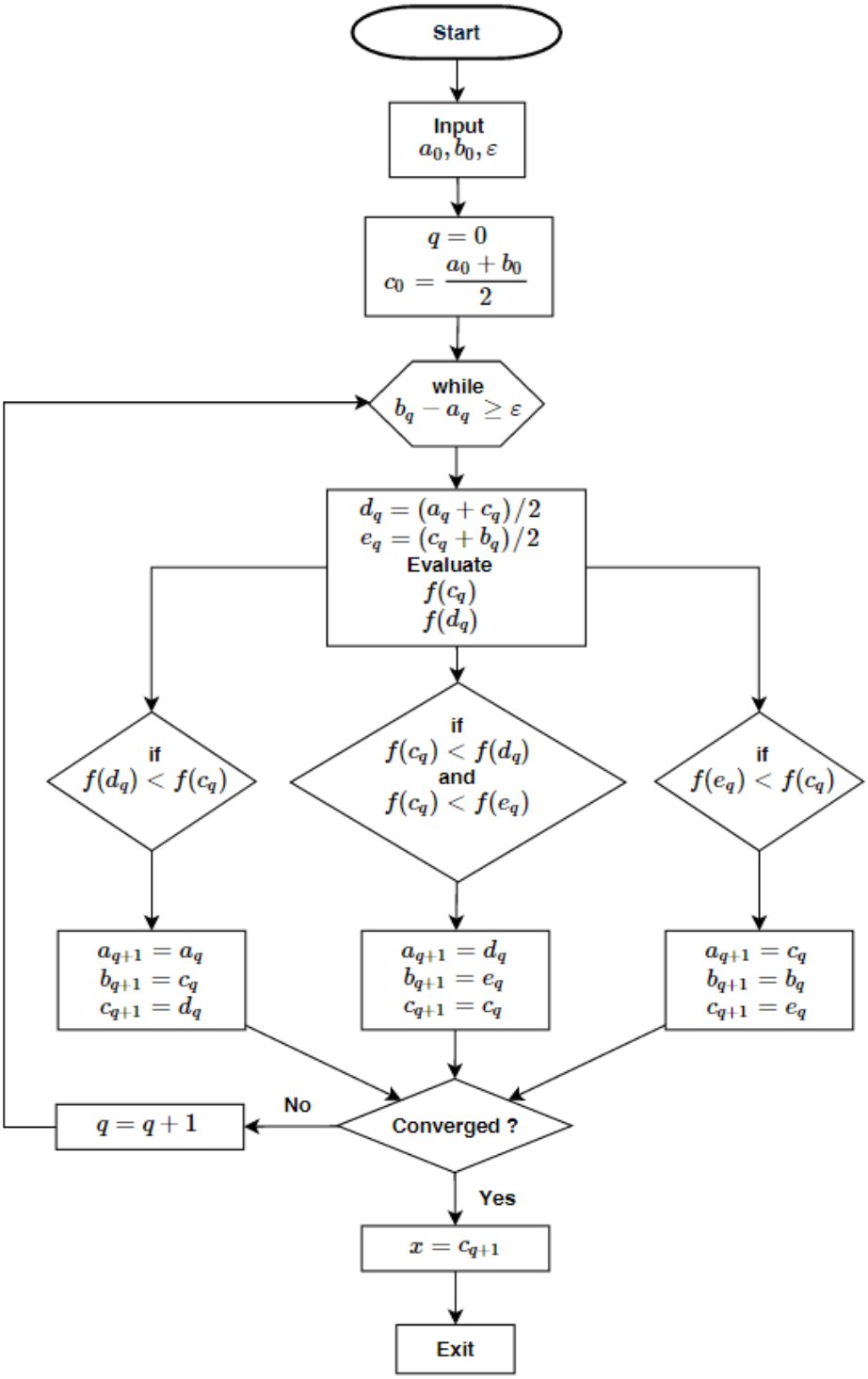


Fig. 25 - Bisection algorithm for unconstrained minimum

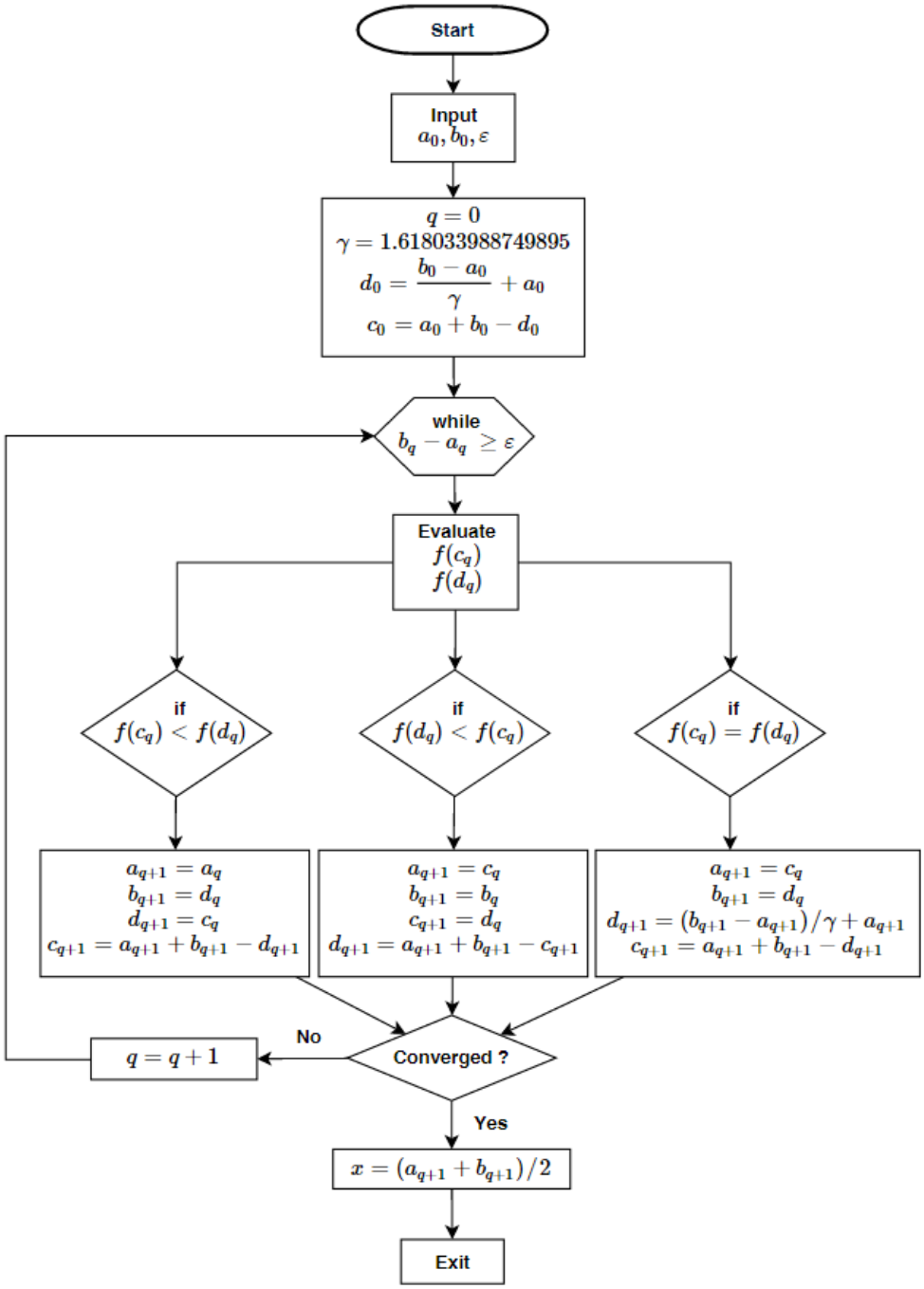


Fig. 26 - Golden section algorithm for unconstrained minimum

With a_0, b_0 the lower and upper bound on step length and ε being the precision.

3.5 Unconstrained optimization methods

Many engineering problems require, that we find the minimum of some function $f(\boldsymbol{x})$, where no restrictions are imposed on the variables $\boldsymbol{x}$. For example nonlinear structural analysis, whereby the displacement field of the structure under given loads is found by minimizing the total potential energy. From the field of tension structures, the form finding or flattening processes may be done using unconstrained optimization methods. Important reason for studying these techniques is that they provide a logical building block toward the solution of nonlinear constrained problems encountered in the design. Especially when sequential unconstrained minimization techniques (SUMT) are used for constrained optimization. This chapter provides several most famous algorithms for unconstrained optimization.

3.5.1 Optimality conditions – unconstrained minimization

For unconstrained optimization problems, local (and global) minima are located at stationary points $\boldsymbol{x}^*$ (asteriks * means the optimal set of $\boldsymbol{x}$). Stationary points are those, for which the gradient of $f(\boldsymbol{x})$ is zero:

$$\nabla f\left(\boldsymbol{x}\right)=\frac{\partial f\left(\boldsymbol{x}\right)}{\partial \boldsymbol{x}}=\mathbf{0} \tag{135}$$

Here, convergence is indicated if each component of $\nabla f(\boldsymbol{x})$ is less then in magnitude than a specified constant ε_1. The designer have to be aware, that the stationary point can also be a local maximum and therefore the decrease of objective function during the optimization process have to be carefully monitored. Other most often used convergence criteria are:

- absolute change in objective function
$$\left|f\left(\boldsymbol{x}^{q}\right)-f\left(\boldsymbol{x}^{q-1}\right)\right|\leq\varepsilon_2 \tag{136}$$
- relative change in objective function
$$\frac{\left|f\left(\boldsymbol{x}^{q}\right)-f\left(\boldsymbol{x}^{q-1}\right)\right|}{\max\left[\left|f\left(\boldsymbol{x}^{q}\right)\right|,10^{-5}\right]}\leq\varepsilon_3 \tag{137}$$
- change in the design
$$\Delta\boldsymbol{x}^{T}\Delta\boldsymbol{x}\leq\varepsilon_4 \tag{138}$$
- number of iterations

where $\Delta\boldsymbol{x}=\boldsymbol{x}^{q-1}-\boldsymbol{x}^{q}$ and ε_i being the convergence criteria with values recommended by [27]: $\varepsilon_1=\varepsilon_3=\varepsilon_4=0.001$ and $\varepsilon_2=0.0001$ or $\varepsilon_2=0.001|f(\boldsymbol{x}^0)|$.

3.5.2 Steepest descent method

The steepest descent method is probably the best known and most easy to understand of all first order methods. The principal importance of the method is that it usually forms the starting point for more sophisticated methods. At each iteration, the search direction $\boldsymbol{S}$ is taken as negative of the gradient of the objective function. That is, at iteration q:

$$\boldsymbol{S}^q = -\nabla f\left(\boldsymbol{x}^{q-1}\right) \tag{139}$$

This search direction is used to perform one dimensional search for optimal step length. When the optimal step length is obtained, the design variables $\boldsymbol{x}^{q-1}$ are updated with $\Delta\boldsymbol{x}^q = \alpha\,\boldsymbol{S}^q$ and the process is iterativelly repeated, until the convergence criteria from section 3.5.1 are satisfied.

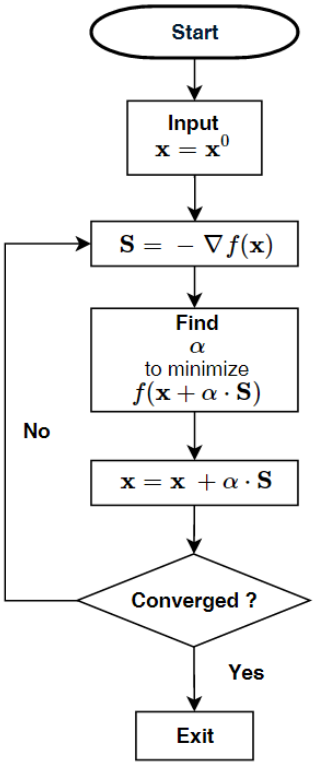


Fig. 27 – Algorithm for the steepest descent method

Since the convergence rate of this algoritm is poor (as will be shown in section 3.5.6), this method is not recommended for general application, although can be used as an initial search direction for more powerfull methods.

3.5.3 Conjugate gradient method

The conjugate gradient method requires only a simple modification to the steepest descent algorithm and yet dramatically improves the convergence rate of the optimization process. The idea is based on constructing the search direction $\boldsymbol{S}$ from more then one gradients, as follows:

$$\boldsymbol{S}^q = -\nabla f\left(\boldsymbol{x}^{q-1}\right) + \beta_q \boldsymbol{S}^{q-1} \tag{140}$$

and the value β depends on the strategy used. For example:

- Fletcher-Reeves strategy

$$\beta_q = \frac{\nabla f\left(\boldsymbol{x}^q\right)^T \nabla f\left(\boldsymbol{x}^q\right)}{\nabla f\left(\boldsymbol{x}^{q-1}\right)^T \nabla f\left(\boldsymbol{x}^{q-1}\right)} \tag{141}$$

- Polak-Ribiere strategy

$$\beta_q = \frac{\nabla f\left(\boldsymbol{x}^q\right)^T \left(\nabla f\left(\boldsymbol{x}^q\right) - \nabla f\left(\boldsymbol{x}^{q-1}\right)\right)}{\nabla f\left(\boldsymbol{x}^{q-1}\right)^T \nabla f\left(\boldsymbol{x}^{q-1}\right)} \tag{142}$$

The conjugate gradient method will theoretically minimize the quadratic function in *n* or fewer iterations. In practice, the optimization problems are not always quadratic and therefore a conjugate search direction, which don´t reduce the value of objective function can be calculated. This can be algorhitmically treated by computing the slope of search direction using dot product of $\boldsymbol{S}^T \nabla f(\boldsymbol{x})$. If its value is positive, the search direction increases the objective function and therefore the search direction is set to the negative of gradient of original function $-\nabla f(\boldsymbol{x})$ (steepest descent). If a steepest descent move still fails to improve the objective, this indicates that the minimum has been found.

While this method is not considered to be as powerful as the second order methods (methods that require second order derivatives of the problem), it has a principal advantage that it is easy to implement and greatly reduces the deficiences of steepest descent method. Basic algorithm for conjugate gradient method is shown on Fig. 28.

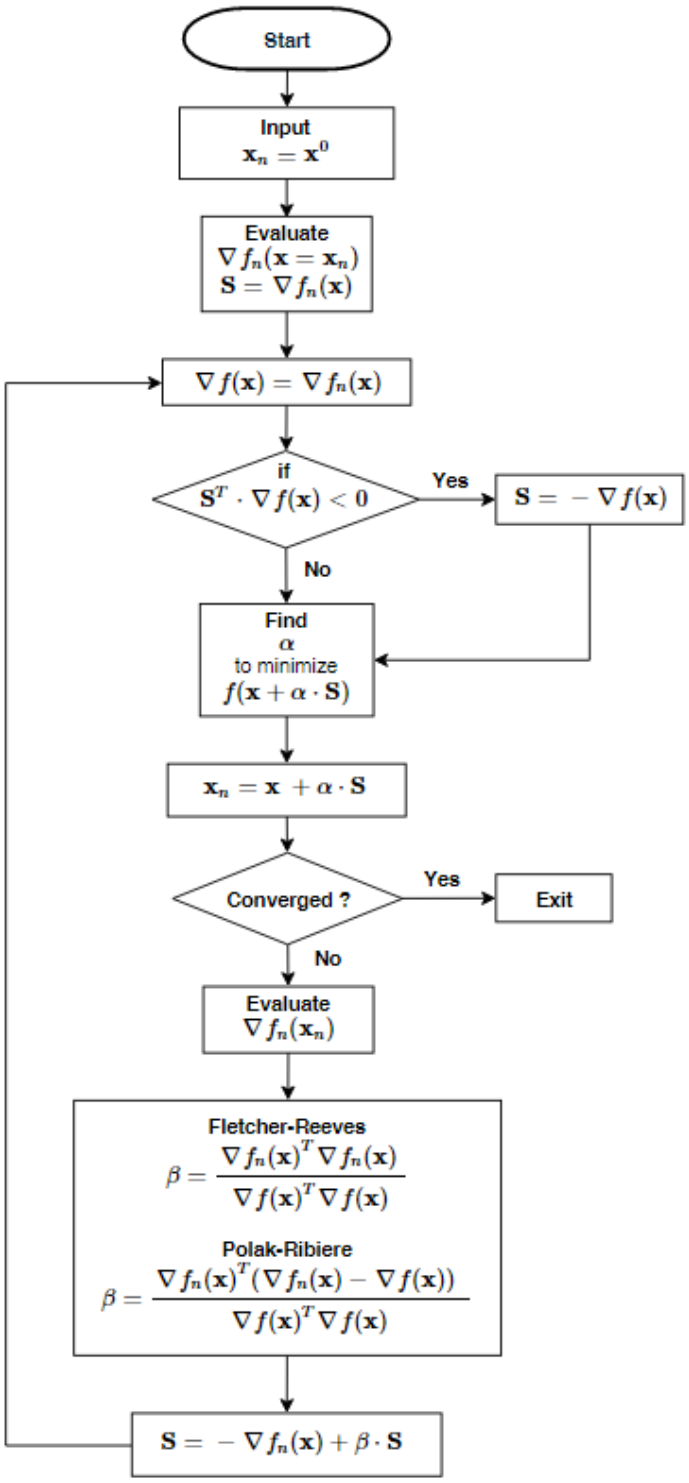


Fig. 28 - Algorithm for conjugate gradient method

3.5.4 Newton´s method

Newton´s method is based on quadratic approximation of an original objective function. The original function $f(\boldsymbol{x})$ is extended by Taylor´s expansion to form:

$$f(\boldsymbol{x}) \cong m(\boldsymbol{x}) = f(\boldsymbol{x}) + \Delta\boldsymbol{x}^T \nabla f(\boldsymbol{x}) + \frac{1}{2}\Delta\boldsymbol{x}^T \nabla^2 f(\boldsymbol{x}) \Delta\boldsymbol{x} \quad (143)$$

minimum is obtained when $\nabla m(\boldsymbol{x}) = \mathbf{0}$, where $\nabla m(\boldsymbol{x}) = \nabla f(\boldsymbol{x}) + \nabla^2 f(\boldsymbol{x}) \Delta\boldsymbol{x}$, from which:

$$\Delta\boldsymbol{x} = -\nabla^2 f(\boldsymbol{x})^{-1} \nabla f(\boldsymbol{x}) \quad (144)$$

and finally $\boldsymbol{x} = \boldsymbol{x} + \Delta\boldsymbol{x}$. This process is repeated until the convergence criteria are met.

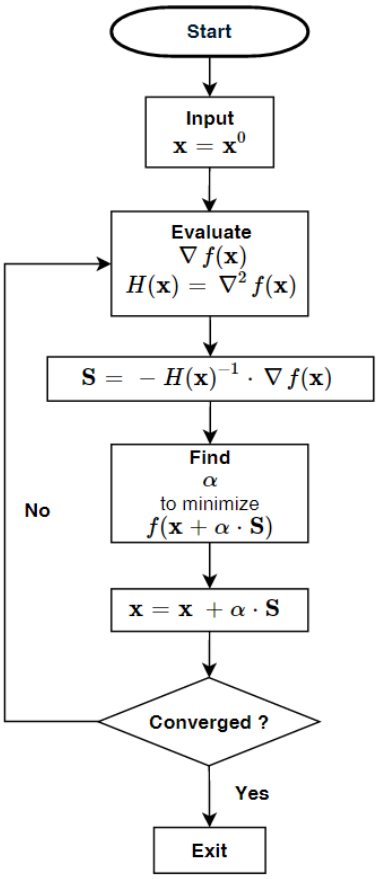


Fig. 29 - Algorithm for full Newton´s method

3.5.5 BFGS Quasi-Newton method

BFGS Quasi-Newton method is one of the best performing variable metric methods. It is based on the same idea as full Newton´s method, with an exception that full Hessian matrix is replaced by its positive definite approximation. This is done mainly because the calculation of full Hessian matrix is very expensive.

At first iteration the identity matrix is used as a Hessian and therefore this iteration is the same as in steepest descent method. Hessian matrix is then iteratively updated with emphasis on its positive definiteness. From the finite difference formula for derivative approximations we can write:

$$f''(\boldsymbol{x}) = \frac{f'(\boldsymbol{x}+\Delta\boldsymbol{x}) - f'(\boldsymbol{x})}{\Delta\boldsymbol{x}} \tag{145}$$

or

$$f''(\boldsymbol{x})\Delta\boldsymbol{x} = f'(\boldsymbol{x}+\Delta\boldsymbol{x}) - f'(\boldsymbol{x}) \tag{146}$$

what can be rewritten into usual optimization notation as:

$$H(\boldsymbol{x})\Delta\boldsymbol{x} = y_c \tag{147}$$

where $H(\boldsymbol{x})$ is the matrix of second derivatives (Hessian), $\Delta\boldsymbol{x} = \boldsymbol{s}_c = \boldsymbol{x}^q - \boldsymbol{x}^{q-1}$ and $\boldsymbol{y}_c = \nabla f(\boldsymbol{x}^q) - \nabla f(\boldsymbol{x}^{q-1})$. Using these variables, the formula for updating Hessian matrix is given as:

$$H(\boldsymbol{x}^q) = H(\boldsymbol{x}^{q-1}) + \frac{\boldsymbol{y}_c \boldsymbol{y}_c^{\,T}}{\boldsymbol{y}_c^{\,T}\boldsymbol{s}_c} - \frac{H(\boldsymbol{x}^{q-1})\boldsymbol{s}_c \boldsymbol{s}_c^{\,T} H(\boldsymbol{x}^{q-1})}{\boldsymbol{s}_c^{\,T} H(\boldsymbol{x}^{q-1})\boldsymbol{s}_c} \tag{148}$$

Check of the positive definiteness of Hessian matrix is done at the end of each iteration. The dot product $\boldsymbol{y}_c^T \boldsymbol{s}_c$ can be rewritten as $\boldsymbol{y}_c^T \boldsymbol{s}_c = \boldsymbol{s}_c^T H^T \boldsymbol{s}_c = \boldsymbol{s}_c^T H \boldsymbol{s}_c$. And if the final product $\boldsymbol{s}_c^T H \boldsymbol{s}_c > 0$, then the Hessian matrix is positive definite and ensures reduction of objective function. If $\boldsymbol{y}_c^T \boldsymbol{s}_c \leq 0$, then no update to Hessian is performed and the last known positive definite approximation is used for the next iteration. Full algorithm of BFGS method is shown on Fig. 30.

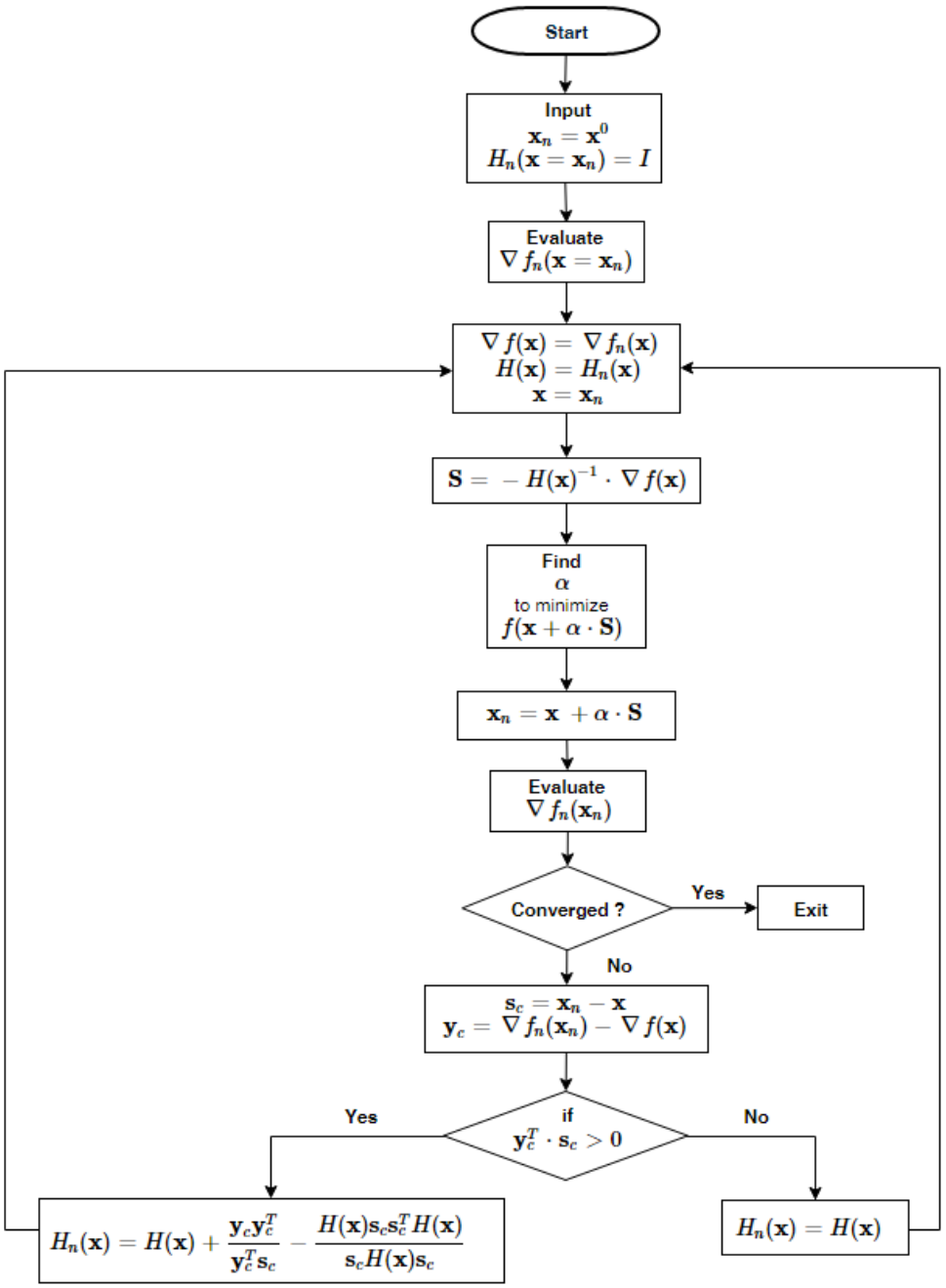


Fig. 30 - Algorithm for BFGS method of unconstrained minimization

3.5.6 Comparison of methods for unconstrained minimization

To provide a comparison, an example function is taken, and all presented methods are tested to show their performance. The objective function is:

$$\min f(x,y) = \sin\left(\frac{x}{3}\right)^2 + \sin\left(\frac{y}{3}\right)^2 \tag{149}$$

with starting point $x = 4$, and $y = 3$. On the following figures, the results from iterations are shown graphically:

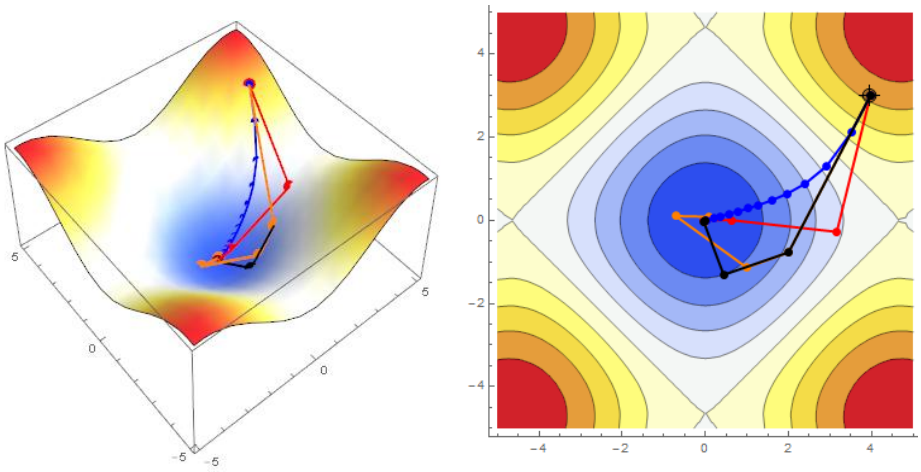


Fig. 31 - Performance of the unconstrained optimization algorithms

The results of individual algorithms are presented in the following table:

Algorithm	Color	Number of iterations
Full Newton´s method		4
BFGS method		6
Conjugated gradient method		4
Steepest descent method		12

Local minimum is at point (0,0) and all methods succeeded in finding it. On this simple function, the superiority of second order methods can be presented as well as very good convergence rate of conjugate gradient method.

3.6 Constrained optimization methods

Most of the real world applications, especially when dealing with shape optimal design, can not be formulated as unconstrained minimization problems. Some limits on sizes, stresses, displacements, etc. have to be prescribed. These limits are called constraints and they divide the design space into feasible domain, where all constraints are satisfied and infeasible domain, where at least one constraint is violated. In most practical problems, the minimum is found on the boundary between these domains. On the boundary is at least one constraint $g(\boldsymbol{x}) = 0$. Other constraints with $g(\boldsymbol{x}) < 0$ can be removed without altering the solution.

The methods which are used to solve constrained optimization problems can be divided into direct and indirect methods. Direct methods deal with the optimization problem as it is formulated by Eqs. (117)-(120). Most typical representatives of direct methods are:

- Linear programming
- Method of feasible directions
- Generalized reduced gradient
- Sequential quadratic programming

Theory behind these algorithms is very well defined, but is beyond the scope of this work. Interested reader is encouraged to study [23], [24], [26], [28], [38] or [39].

On the other hand, the indirect methods convert the original objective function to a sequence of equivalent unconstrained minimization problems. These methods are then called sequential unconstrained minimization techniques (SUMT) and this work is focused on them.

The classical approach to using SUMT is to create pseudo-objective function of the form:

$$\phi\left(\boldsymbol{x}, r_p\right) = f\left(\boldsymbol{x}\right) + r_p P\left(\boldsymbol{x}\right) \tag{150}$$

where $f(\boldsymbol{x})$ is the original objective function, $P(\boldsymbol{x})$ is an imposed penalty function and r_p is a multiplier, which determines the magnitude of the penalty. Depending on construction of penalty function $P(\boldsymbol{x})$ we differ the type of SUMT algorithm used (see sections 3.6.2, 3.6.3 and 3.6.4). This pseudo-objective is then sequentialy minimized by unconstrained optimization techniques described in chapter 3.5.

3.6.1 Optimality conditions – constrained minimization

In order to check whether the constrained optimization algorithm converged to at least local minimum, the optimality conditions presented is section 3.5.1 are necessary, but not sufficient. Optimal solution have to meet also other requirements, which are collected into so called Karush-Kuhn-Tucker conditions (KKT). Before the KKT conditions are descibed, the Lagrangian function of constrained problem is introduced:

$$L(\boldsymbol{x},\boldsymbol{\lambda}) = f(\boldsymbol{x}) + \sum_{j=1}^{m} \lambda_j g_j(\boldsymbol{x}) + \sum_{k=1}^{l} \lambda_{m+k} h_k(\boldsymbol{x}) \tag{151}$$

Lagrangian function extends the objective function with the vectors of constraints $g_j(\boldsymbol{x})$, $h_k(\boldsymbol{x})$ and their corresponding Lagrange multipliers λ_j and λ_{m+k} respectivelly. Lagrange multipliers are often reffered to as „slack variables“ which define the sensitivity of optimum with respect to relative change in the constraint bounds. KKT conditions define a stationary point of the Lagrangian and state simply that if vector $\boldsymbol{x}^*$ defines the optimum design, the following conditions must be satisfied:

1. Stationarity

$$\nabla f(\boldsymbol{x}^*) + \sum_{j=1}^{m} \lambda_j \nabla g_j(\boldsymbol{x}^*) + \sum_{k=1}^{l} \lambda_{m+k} \nabla h_k(\boldsymbol{x}^*) = \mathbf{0} \tag{152}$$

2. Primal feasibility

$$g_j(\boldsymbol{x}^*) \le 0 \quad \text{for} \quad j = 1 \ldots m \tag{153}$$

$$h_k(\boldsymbol{x}^*) = 0 \quad \text{for} \quad k = 1 \ldots l \tag{154}$$

3. Dual feasibility

$$\lambda_j \ge 0 \tag{155}$$

$$\lambda_{m+k} \text{ - unrestricted in sign} \tag{156}$$

4. Complementary slackness

$$\lambda_j \nabla g_j(\boldsymbol{x}^*) = 0 \quad \text{for} \quad j = 1 \ldots m \tag{157}$$

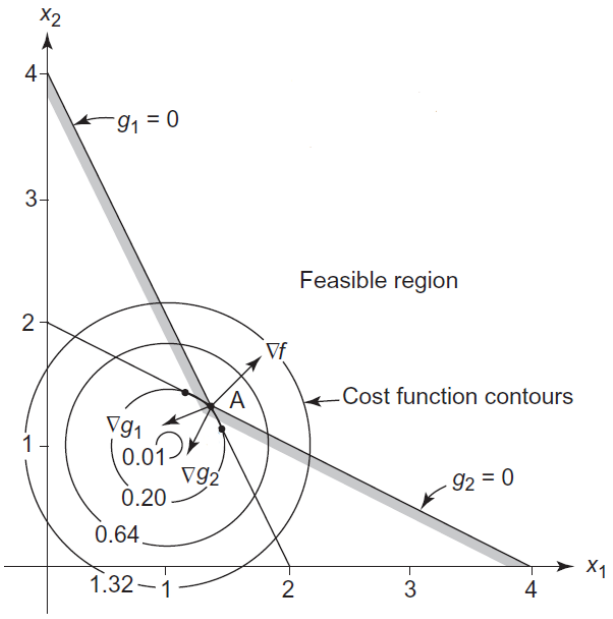


Fig. 32 - Graphical representation of KKT conditions

Stationarity condition requires, that at the optimum $\boldsymbol{x}^*$, the gradient of Lagrangian must vanish. When no constraints are present, then the gradient of Lagrangian is equal to gradient of original objective function and therefore the stationarity condition is equal to condition given in section 3.5.1. Primal feasibility conditions require that, the design at the optimum must be feasible, e.g. all constraints and bounds on variables must be satisfied. Dual feasibility and complementery slackness impose the requirement, that if the inequality constraint $g_j(\boldsymbol{x}^*)$ is not precisely satisfied /that is, $g_j(\boldsymbol{x}^*) < 0$/, then the corresponding Lagrange multiplier λ_j must be zero. For preciselly satisfied inequality conditions and all equality conditions the corresponding Lagrange mulipliers can have any value (or any positive value in case of inequality conditions).

If the design space is convex, the KKT conditions define global optimum, otherwise only a relative (local) optimum is found. For a convex design space, the Hessian matrix of objective function and all constraints must be positive definite for all possible combinations of the design variables. This can seldom be demonstrated in practical applications and hence we must usually be satisfied with starting the design from various initial points to see, if we can obtain a consistent optimum.

3.6.2 Penalty function method

Penalty method, or often stated as the exterior penalty function method is the easiest method to implement into optimization process. Penalty function $P(\boldsymbol{x})$ is typically given by:

$$P(\boldsymbol{x}) = \sum_{j=1}^{m} \left\{ \max\left[0, g_j(\boldsymbol{x})\right] \right\}^2 + \sum_{k=1}^{l} \left[h_k(\boldsymbol{x}) \right]^2 \tag{158}$$

From Eq. (158) can be seen, that if no constraints are violated /all $g_j(\boldsymbol{x}) < 0$ and $h_k(\boldsymbol{x}) = 0$/, then penalty term is equal to zero and optimization process procceds like unconstrained. Penalty resulting from constraint violation is added to original objective to form pseudo-objective function:

$$\phi(\boldsymbol{x}, r_p) = f(\boldsymbol{x}) + r_p P(\boldsymbol{x}) \tag{159}$$

and this is then minimized using first-order unconstrained optimization methods (steepest descent or conjugate gradient methods). Second-order methods are generally not recommended because second derivatives of penalty function are not continuous and the numerical ill-conditioning may appear.

Penalty multiplier r_p is used to control the behaviour of the method. If we choose smal value for r_p, the resulting function $\emptyset(\boldsymbol{x}, r_p)$ is easily minimized, but may yieald in large constraint violation. On the other hand, a large value of r_p will ensure near satisfaction of all constraints, but can cause numerical difficulties. Therefore, usually small value for r_p is chosen at the beginning of the optimization process and in subsequent iterations is increased by factor γ ($\gamma = 5$ and 3 is recommended in literature [23] and [27] respectivelly). The general algorithm for penalty method is shown on Fig. 33.

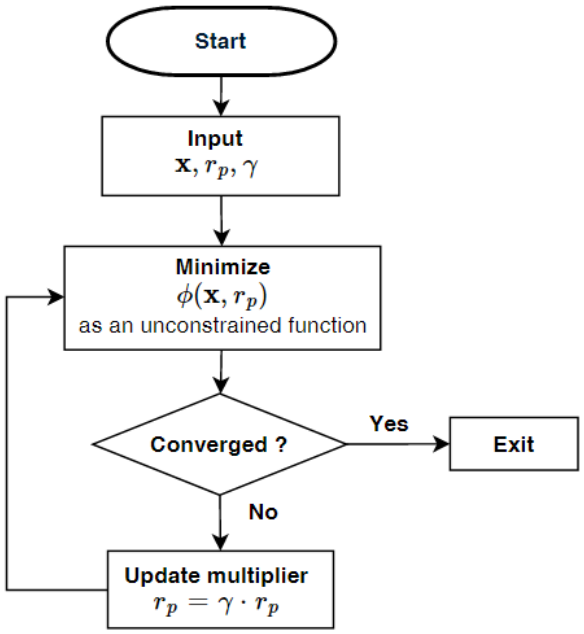


Fig. 33 - Algorithm for (exterior) penalty function method

3.6.3 Barrier method

The main disadvantage of the exterior penalty function is that, it is defined only in the infeasible domain [23]. When only inequality constraints are present, it is possible to define an interior penalty function that keeps the design in the feasible domain. The most common forms of the interior penalty function are:

$$P(\boldsymbol{x}) = \sum_{j=1}^{m} -\frac{1}{g_j(\boldsymbol{x})} \tag{160}$$

or an alternative form, slightly better numerically conditioned:

$$P(\boldsymbol{x}) = \sum_{j=1}^{m} -\log\left[-g_j(\boldsymbol{x})\right] \tag{161}$$

When $g_j(\boldsymbol{x})$ approaches zero, the penalty term $P(\boldsymbol{x})$ in both presented cases becomes infinitelly large and therefore creating a barrier between feasible and infeasible domain of design space (thus the name of the method). The penalty paramater r_p' used in barrier (or interior penalty function) method, unlike in penalty function method, begins as a relatively large positive

number and decreases in subsequent iterations by factor γ' ($\gamma' = 0.3$ is recommended in literature [27]). This ensures that in situation, when the inequality constraints are precisely satisfied $/g_j(\boldsymbol{x}) = 0/$, the penalty multiplier r_p' approaches zero and therefore cancels the impact of infinitelly large penaly function $P(\boldsymbol{x})$.

When the designer needs to include also equality constriants, the pseudo-objective function of barrier method (for inequality constraints) is extended with the penalty term from penalty method. The final pseudo-objective is then:

$$\phi\left(\boldsymbol{x}, r_p, r_p'\right) = f\left(\boldsymbol{x}\right) + r_p' \sum_{j=1}^{m} -\frac{1}{g_j\left(\boldsymbol{x}\right)} + r_p \sum_{k=1}^{l} \left[h_k\left(\boldsymbol{x}\right)\right]^2 \qquad (162)$$

It is important to distinguish the difference between r_p and r_p'. While the r_p from penalty function method starts as initially small value and increases, the penalty multiplier r_p' from barrier method starts as initially large number an decreases. The full algorithm is shown on Fig. 34.

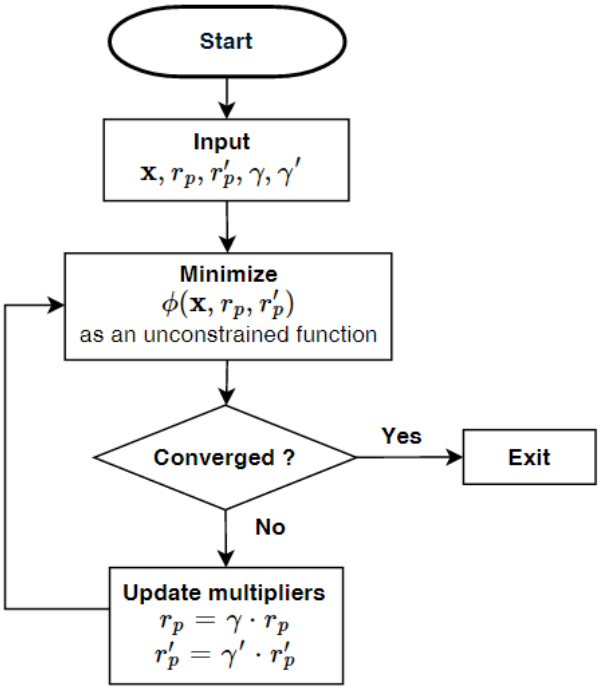


Fig. 34 - Algorithm for barrier (interior penalty function) method

3.6.4 Augmented Lagrange Multiplier method

This method is based on unconstrained minimization of pseudo-objective, which is created as a combination of exterior penalty method and Lagrange function. Resulting pseudo-objective function takes the form:

$$L\left(\boldsymbol{x},\lambda,r_p\right)=f\left(\boldsymbol{x}\right)+\sum_{j=1}^{m}\left[\lambda_j\psi_j+r_p\psi_j^2\right]+\sum_{k=1}^{l}\left[\lambda_{m+k}h_k\left(\boldsymbol{x}\right)+r_p\left[h_k\left(\boldsymbol{x}\right)\right]^2\right] \tag{163}$$

where

$$\psi_j=\max\left[g_j\left(\boldsymbol{x}\right),\frac{-\lambda_j}{2r_p}\right] \tag{164}$$

After the convergence of unconstrained minimization, the Lagrange and penalty multipliers and have to be updated. The update formula of multipliers for inequality constraints is:

$$\lambda_j^{p+1}=\lambda_j^p+2r_p\left\{\max\left[g_j\left(\boldsymbol{x}\right),\frac{-\lambda_j^p}{2r_p}\right]\right\} \tag{165}$$

for equality constraints:

$$\lambda_{k+m}^{p+1}=\lambda_{k+m}^p+2r_ph_k\left(\boldsymbol{x}\right) \tag{166}$$

and for penalty multipliers:

$$r_p^{p+1}=\gamma r_p^p \tag{167}$$

In the previous formulas, p in superscript means the iteration number and p in subscript stands for „penalty“. Augmented Lagrangian Multiplier method (ALM) is numerically well conditioned and therefore penalty multipliers r_p need not to be increased to theoretical infinity. In practical applications we usually set a maximum value r_p^{max}, which is not exceeded during the optimization procedure. Since we can not specify a suitable value of r_p^{max} in advance, we usually set the maximum relativelly large, for example 10^5. The method is commonly reccomended to begin with $\lambda=0$ and an arbitrary (but small) value for r_p. The algorithm for this method is given on Fig. 35.

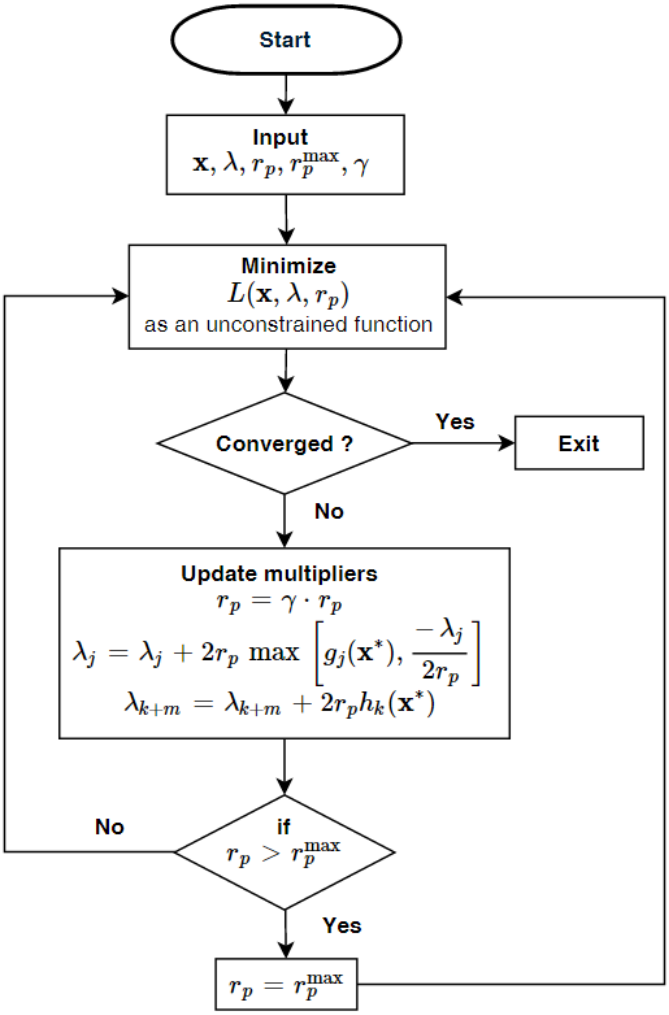


Fig. 35 - Algorithm for general case of Augmented Lagrangian Multiplier method

Augmented Lagrangian multiplier method is superior over other SUMT methods, it is robust and suitable for large scale problems. In summary, the method has several attractive features:

- the method is relatively insensitive to the value of r_p. It is not neccessary to inrease r_p to infinity
- precise $g_j(\boldsymbol{x}) = 0$ and $h_k(\boldsymbol{x}) = 0$ is possible
- the starting point may be either feasible or infeasible
- at the optimum, the value of $\lambda_j^* \neq 0$ will automatically identify the active constraints

3.6.5 Comparison of methods for constrained minimization

As in section 3.5.6, also here an example is given to compare the performence of presented algorithms. Consider a following constrained optimization problem:

$$\min f(x,y) = (1-x)^2 + 100\left(y - x^2\right)^2 \tag{168}$$

subject to

$$g(x,y) = -2 + (-5+x)^2 + y^2 \tag{169}$$

with starting point $x = 4.8$, and $y = -2.7$. On the following figures, the results from iterations are shown graphically:

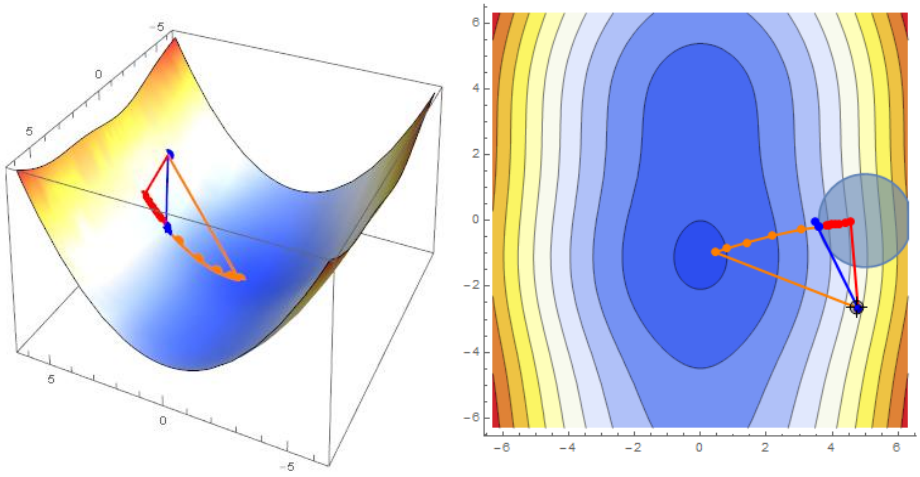


Fig. 36 - Performance of algorithms for constrained optimization

Colored isosurfaces represent the contours of objective function and light-blue circle represent the feasible domain satisfying the constraint. The results of individual algorithms are presented in the following table:

Algorithm	Color	Number of iterations
Penalty method		7
Barrier method		6
Augmented Lagrangian Multiplier method		3

Local minimum (and also constrained global minimum) was found at point (3.6;-0.2) and all algorithms succeeded. The convergence rate of ALM method has proven to be the best.

3.7 Example

The task is to design a simple hypar membrane structure with geometry given on Fig. 37 and analyze it under loads given on Fig. 44. After the usual design process (geometric form finding, mechanical form finding, static analysis), the optimization procedures are applied with aim of reducing fabric stresses. This can be achieved by modifying the membrane and edge cable prestress within given bounds. For the example to be more illustrative, the edge cables are divided into two independent design variables and will be treated separately to demonstrate the basic background of optimization technology. The obtained results are discussed at the end of the example.

Structural design problem

$E_1 = 0{,}9e^6$ GPa
$E_2 = 0{,}648e^6$ GPa
$G_{12} = 0{,}031e^6$ GPa
$\nu_1 = 0{,}018$
$\nu_2 = 0{,}025$
thickness = 0,001 m
warp = X direction
weft = Y direction

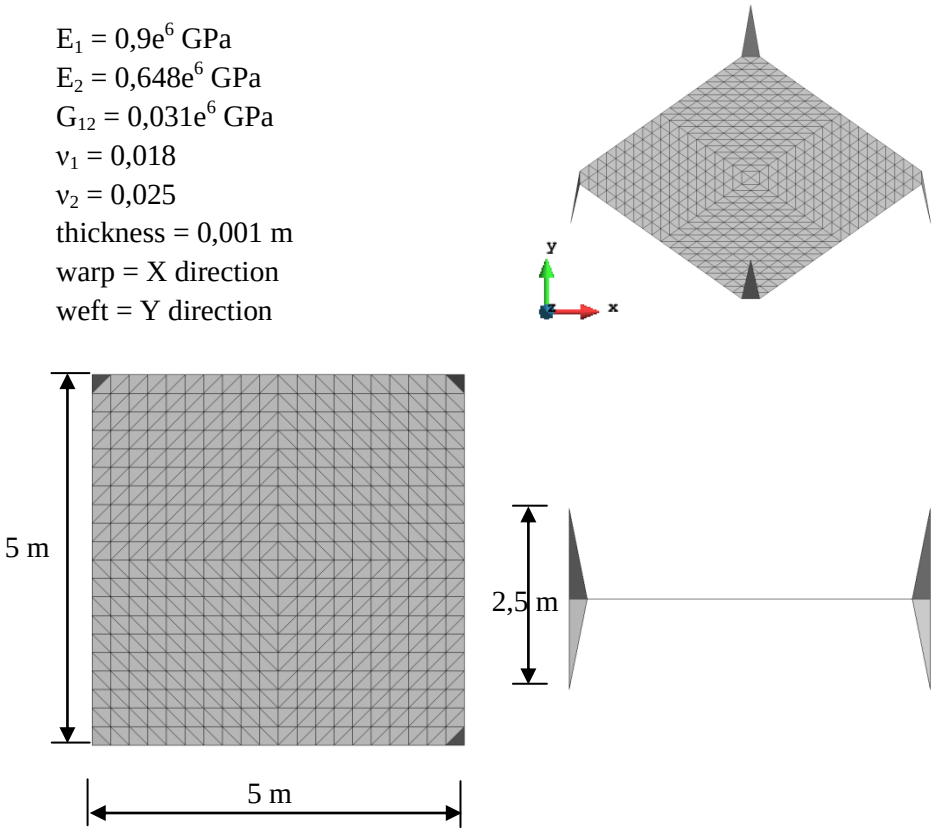
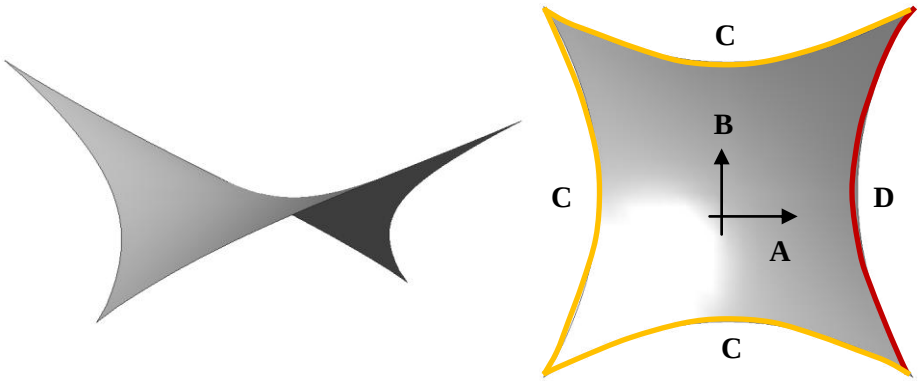


Fig. 37 - Structural problem description



Optimization problem

minimize maximum stress	$\min(\max(\sigma_x, \sigma_y, \tau_{xy}))$	
design variables	warp prestress	A
	weft prestress	B
	yellow cables pretension	C
	red cable pretension	D
limit values	$2 \leq A \leq 6$ kN/m	
	$2 \leq B \leq 6$ kN/m	
	$15 \leq C \leq 60$ kN	
	$15 \leq D \leq 60$ kN	
initial values	A = 5 kN/m	
	B = 5 kN/m	
	C = 50 kN	
	D = 50 kN	

Fig. 38 - Optimization problem description

To solve this optimization problem a medium-scale SQP algorithm (Sequential Quadratic Programming) with BFGS quasi-Newton Hessian approximation and linesearch steplength is used. This algorithm is not presented in this work due to the wide theoretical background, but Augmented Lagrangian Multiplier method presented in section 3.6.4 can be very effectively used [13]. As a termination criterion the absolute change in objective function was used (Eq.(136)), with $\varepsilon_2 = 0.01$. Finite element analysis is limited to 25 full Newton-Raphson iterations and is considered as converged, when Euclidean norm of residual forces *R* is smaller than 0,01.

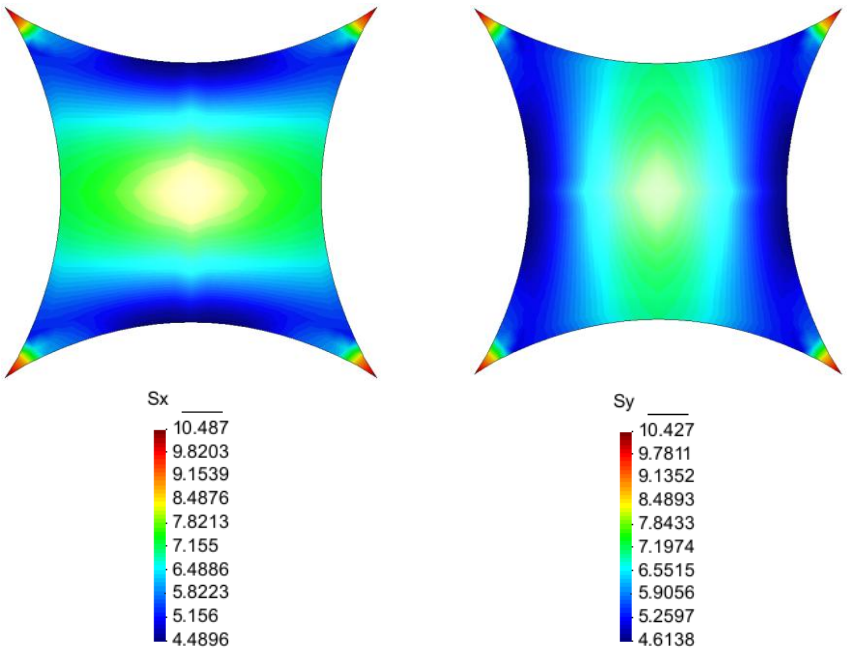


Fig. 39 - Membrane stresses after geometric form finding

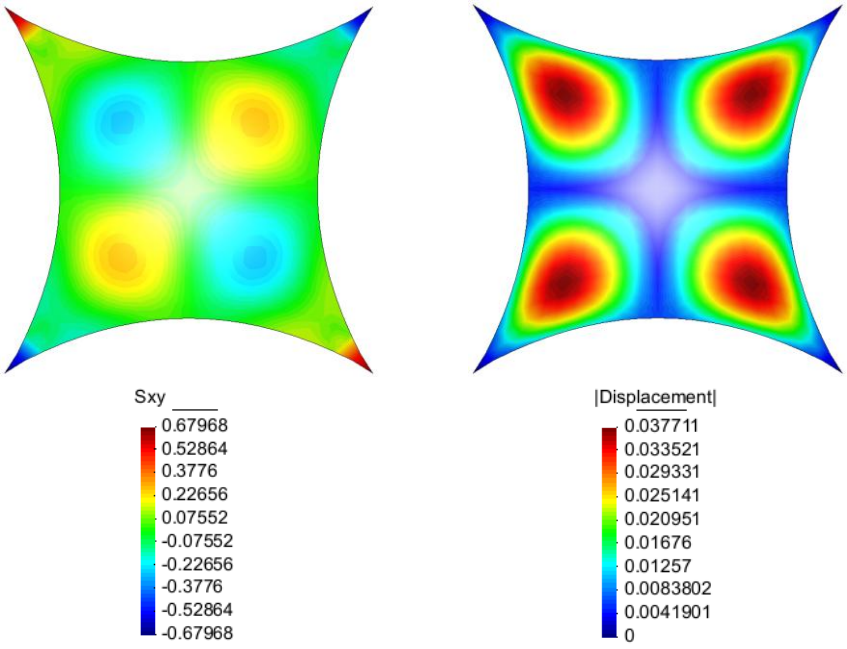


Fig. 40 - Shear stresses and total displacements after geometric form finding

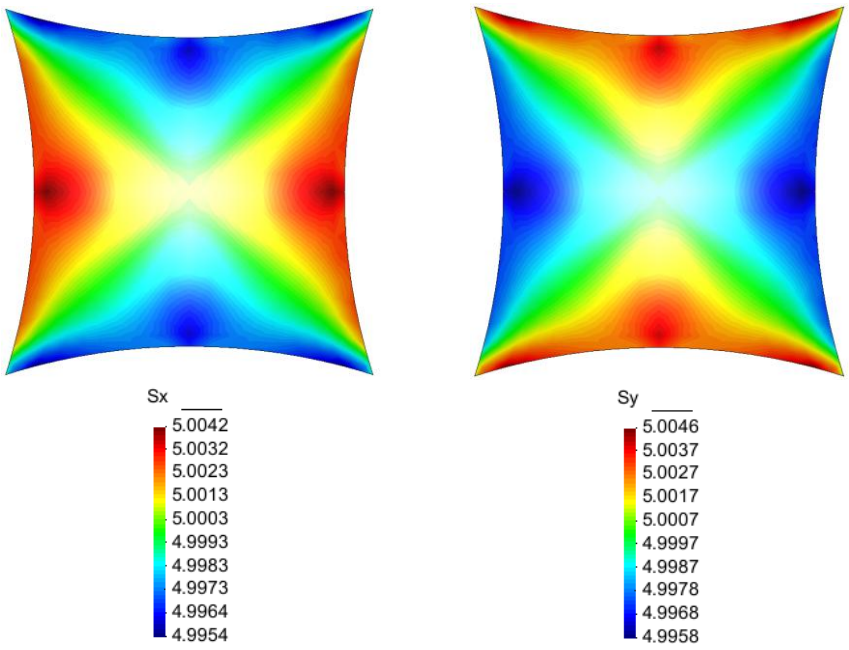


Fig. 41 - Membrane stresses after mechanical form finding

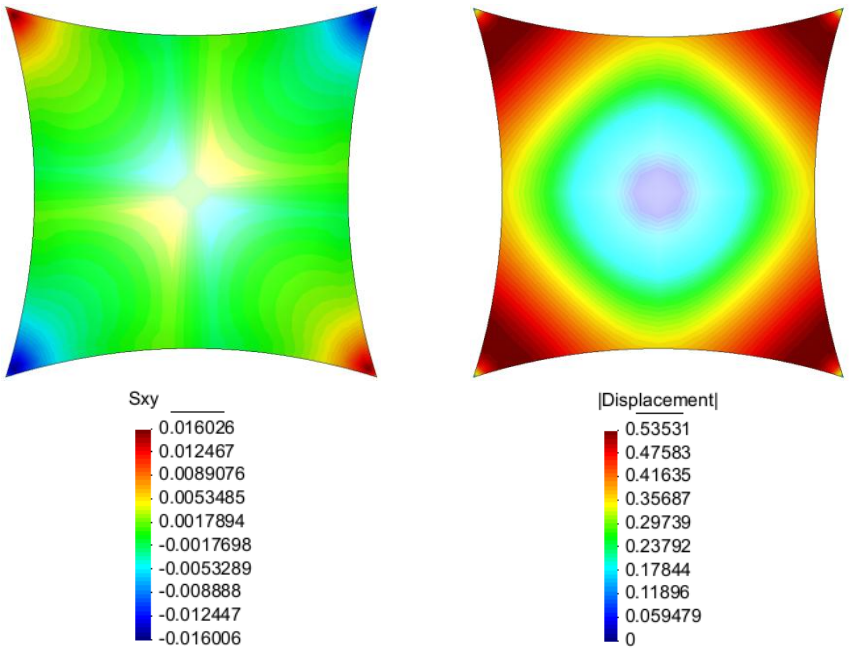


Fig. 42 - Shear stresses and geometrical differences between geometric and mechanical form finding

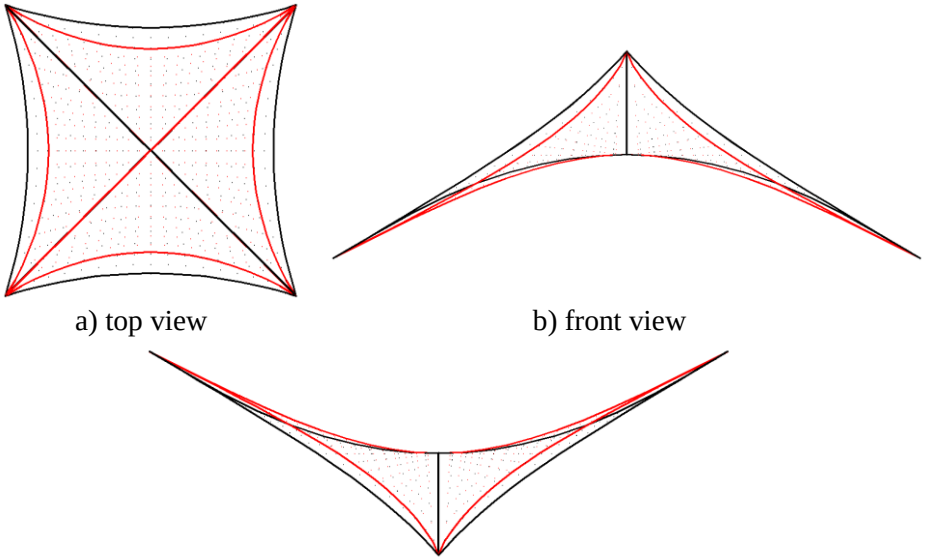


Fig. 43 - Geometrical differences between geometric and mechanical form finding (red - geometric, black - mechanical)

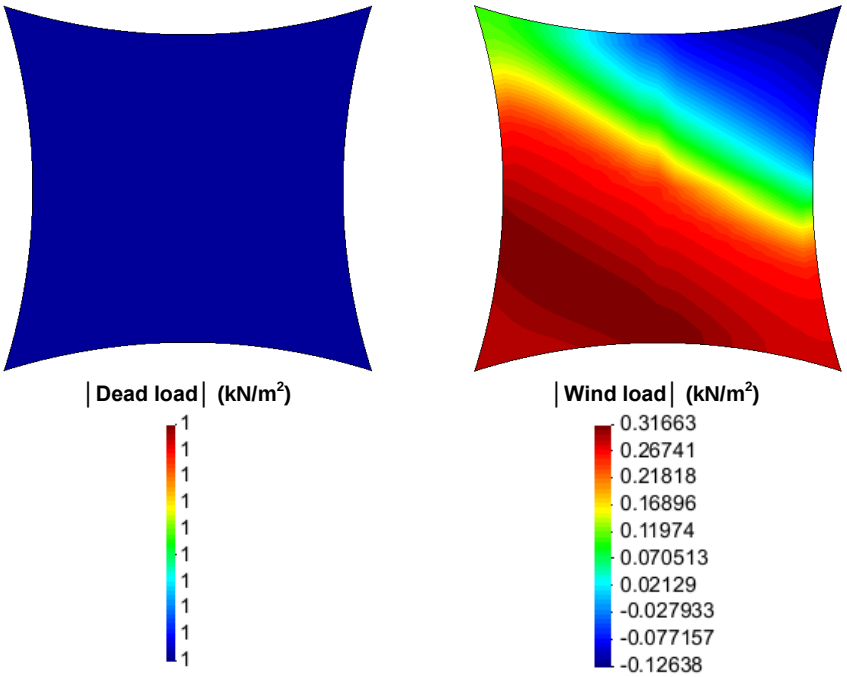


Fig. 44 - Membrane loads

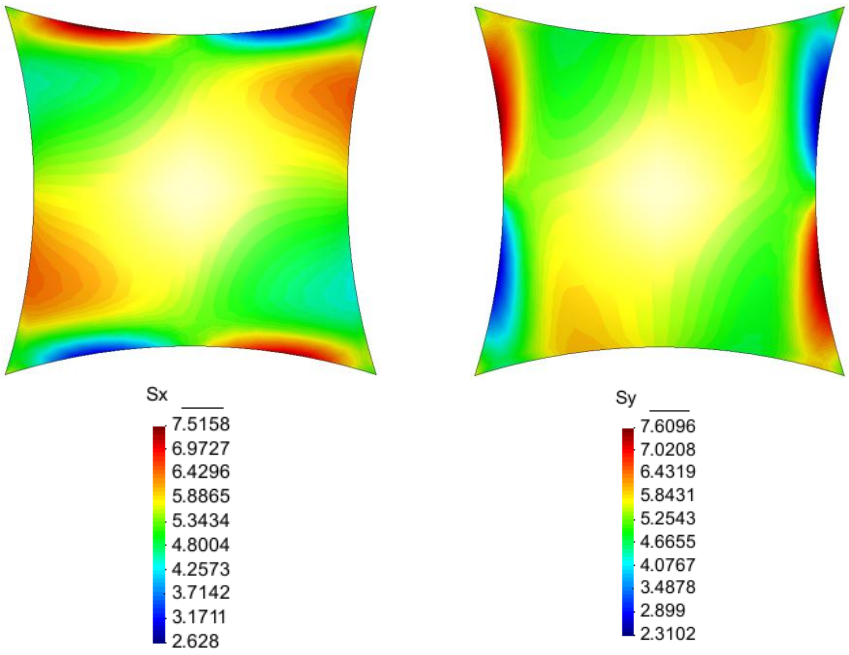


Fig. 45 - Membrane stresses from static analysis

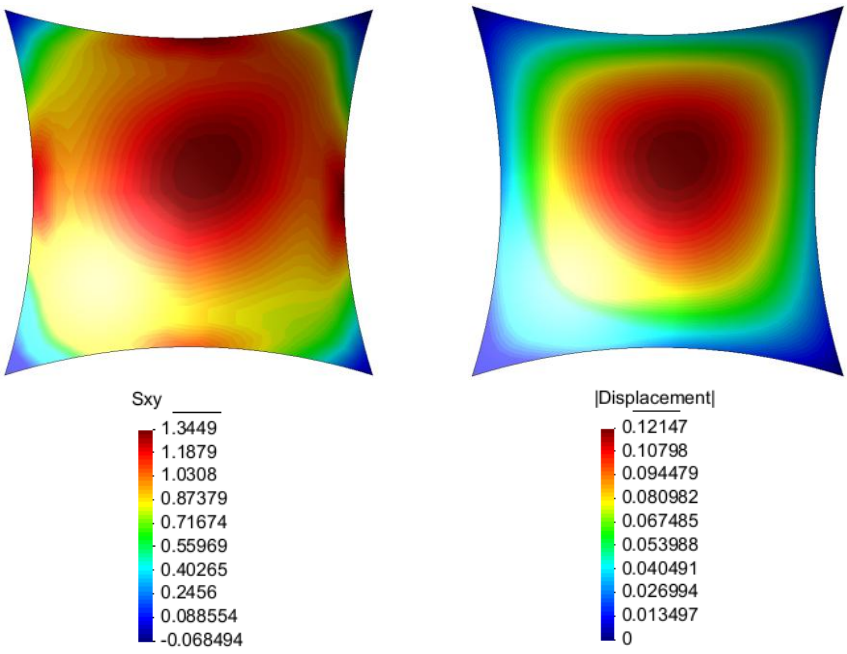


Fig. 46 - Shear stress and displacements from static analysis

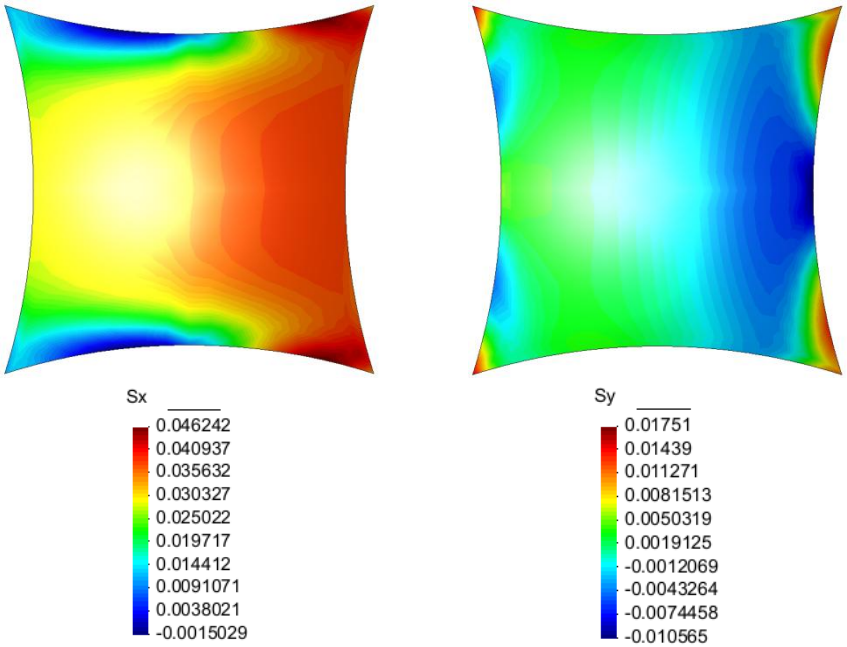


Fig. 47 - Membrane stress sensitivities to increase of pretension of cable D

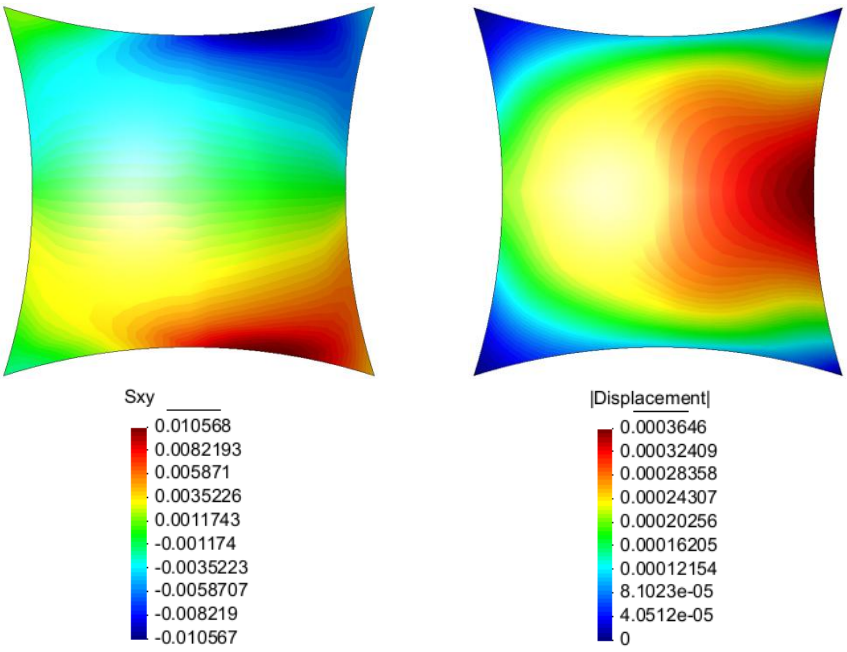


Fig. 48 - Shear stress and total displacement sensitivities to increase of pretension of cable D

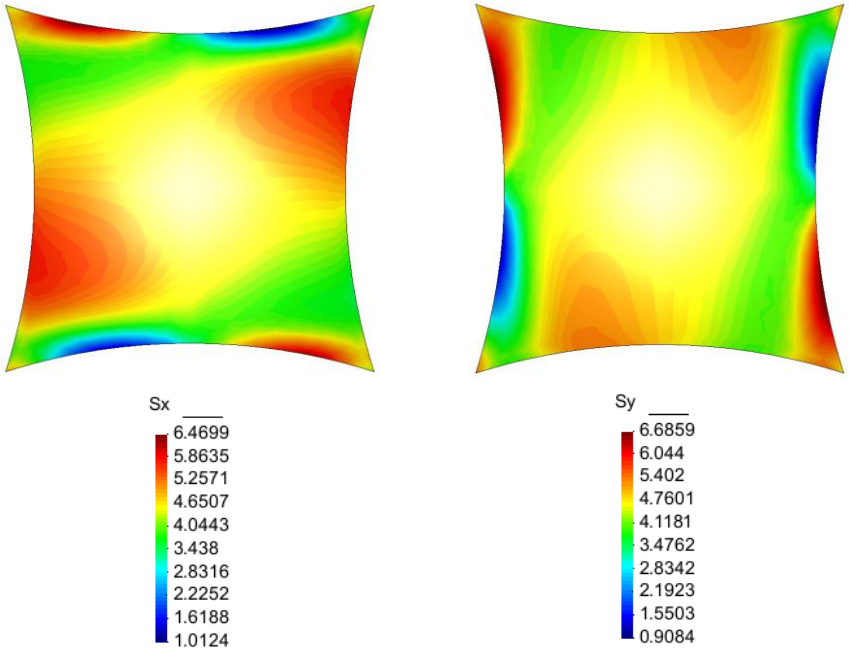


Fig. 49 - Membrane stresses with optimized prestress

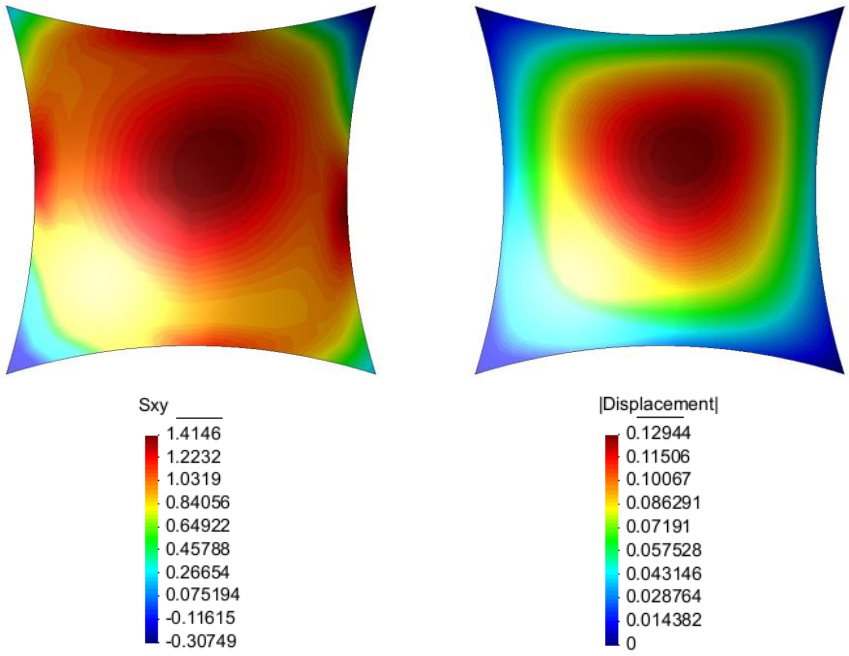
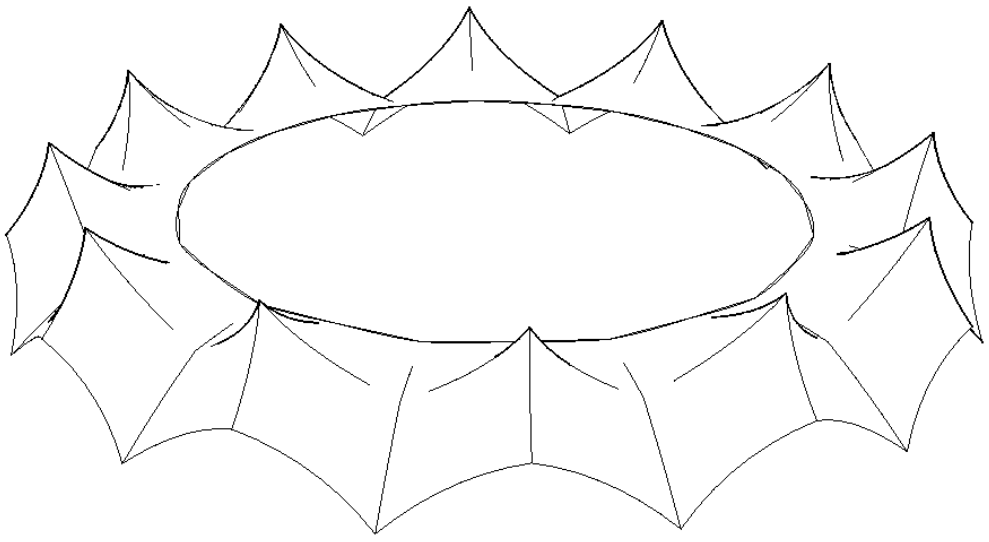


Fig. 50 - Shear stresses and total displacements with optimized prestress

3.7.1 Results and discussion

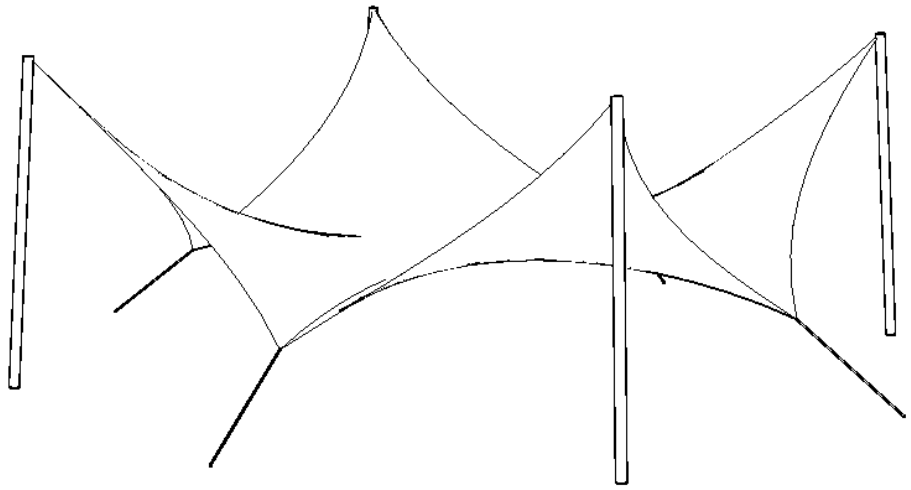
The aim of previous example was to demostrate the practical application of procedures described in this work. Among the most interesting phenomena belongs the difference between geometric and mechanical form finding results. While the geometrical form finding results in a shape on which the direction cosines of surface tangents are in equilibrium, the mechanical form finding takes into account also the magnitude of these surface tangents. Therefore there can be a large differences between both form finding methods. Although the mechanical form finding is not neccessary design step, it is recommended to do it. If it is not done, the out-of-balance nodal forces resulting from prestress will have an influence on nodal displacements in the analysis stage. Although it has no influence on final stress state, the designer can be confused when interpreting the results, because final displacement field will consist of displacements caused by external loads and displacements caused by prestress. As shown on Fig. 47 - Sy, the application of different prestress then used during formfinding, can lead to undesirable reduction of stress, what often results in wrinkled edges.
Fig. 47 and Fig. 48 shows the sensitivity of the structure with respect to change in cable D pretension. Fig. 47 presents the change in the membrane stress state, when the cable D pretension is increased by 1 kN. It can be seen, that the stress in X direction generally increases, but in the close neighborhood of this cable the membrane stresses in Y direction shows decrease. Also the influence on shear stress and total nodal displacements are shown on Fig. 48. These results are intuitive when one understand the concept of introducing pretension. Once the designer is informed about the influence of the unit change of all design variables, the magnitude of this change can be chosen.
In our example, the optimizer found a localy optimal values of design variables (prestress) as follows: A = 2,81; B = 3,04; C = 49,77; D = 50. The initial maximum stresses were σ_x = 7,51 kN/m; σ_y = 7,61 kN/m; τ_{xy} = 1,34 kN/m (Fig. 45, Fig. 46). Stresses with optimized prestress σ_x = 6,47 kN/m; σ_y = 6,69 kN/m; τ_{xy} = 1,41 kN/m (Fig. 49, Fig. 50) shows a decrease of 13,8 % in X direction, 12,1 % in Y direction and increase of 5,2 % in shear stress. The fabric prestress was reduced by 43,6 % in warp direction and 39,1 % in weft direction. Cable pretension remained unchanged, since its influence on final stress state is negligible comparing to influence of fabric prestress. The optimizer also revealed, that the edge cables have equivalent infulence on final prestress (i.e. C = D) and therefore don´t have to be dealt separately. Different final prestress of fabric in warp and weft directions is the result of different material properties in those directions, which was also recognized by optimizer. Since warp direction is stiffer, it requires less prestress.



4 Summary

This work aims to provide on overview and brief insight to numerical background of design and optimization of tensile structures. Deeper knowledge of these techniques gives to designer the possibilities for doing „free experiments“, which would otherwise require skilled handwork or would cost much time and money. Numerical techniques also enlarges the design space. Hybrid structures are such an example. Inclusion of bending active elements (beams or shells) to tensile structures opens a new space of possible shapes and forms. When one looks on form found shape as on shape in equilibrium under given loads, numerical incorporation of other then tensile active elements is clear, straightforward and requests no more energy or wisdom then usual approach. This can be only seldom achieved by physical models, since for the deformation of elements active in bending, the effort of work and consumed energy grows rapidly. Numerical techniques used these days, such as computational mechanics, are very well defined and their idealization of structures offers reliable description of real structural behaviour. Additionally, mechanical approach, described in chapter 2 can be used in every part of the design process of tensile structures. Last, but not least, application of optimization techniques described in chapter 3, if used effectivelly, can greatly reduce engineering design time and yield improved, efficient and economical design.

With the procedures described in this work, the reader will be able to perform necessary calculations required by every stage of the tensile structures design. It should be kept in mind, that the field of computational mechanics and structural optimization is much wider, than described here. Interested reader is encouraged to study the references for deeper understanding of presented procedures.



5 Literature

[1] ADINA R&D – Theory and Modeling Guide, Vol. 1: ADINA Solids & Structures, ADINA R & D Inc., 2012

[2] Bathe, K.J. - Finite element procedures, Second Edition, Prentice Hall, 2014

[3] Crisfield, M.A. - Nonlinear finite element analysis of solids and structures, Vol. 1 and Vol. 2, Wiley, 1997

[4] Oñate, E. - Structural Analysis with the Finite Element Method, Vol. 1 (2009) and Vol 2. (2013), Springer

[5] Przmieniecki, J.S. – Theory of Matrix Structural Analysis, McGraw-Hill Inc., 1968

[6] Rao, S. S. - The Finite Element Method in Engineering, 5th Edition, Elsevier Inc., 2011

[7] Zienkiewicz, O.C., Taylor, R.L., Zhu, J.Z. - The Finite Element Method, its Basis and Fundamentals, Vol. 2, 5th Edition, Elsevier Butterworth-Heinemann, 2000

[8] Forster, B.,Mollaert, M. - European Design Guide for Tensile Surface Structures, Tensinet, 2004

[9] Argyris, J. H., Angelopoulos, T., Bichat, B. - A general method for the shape finding of lightweight tension structures, Comput. Meth. Appl. Mech. Engng 3, 135-149 1974

[10] Tabarrok, B., Qin, Z. - Nonlinear Analysis of Tension Structures, Computers and Structures, Vol. 45, No. S/6, pp. 913-984, 1992

[11] Haber, R.B., Abel, J.F. - Initial equilibrium solution methods for cable reinforced membranes, Part I and II. Computers Methods in Applied Mechanical Engineering, 30, 263–289 et 285–30, 1982

[12] Nouri-Baranger, T. - Form-finding method of tensile fabric structures: revised geometric stiffness method, Journal of the International Association for Shell and Spatial Structures, 43 (1), April N. 138, 13–22, 2002

[13] Nouri-Baranger, T. – Computational methods for Tension-Loaded Structures, Arch. Comput. Meth. Engng. Vol. 11, 2, 143-186, 2004

[14] Adriaenssens, S., Block, P., Veenendaal, D., Williams, C. - Shell structures for architecture: Form finding and optimization, Routledge, 2014

[15] Contri, P., Schrefler, B. A. - A geometrically nonlinear finite element analysis of wrinkled membrane surfaces by a no-compression material model. Commun. Appl. Numer. Meth. 4, 5-15 1988

[16] Topping, B. H. V., Iványi, P. - Computer Aided Design of Cable Membrane Structures, Saxe-Coburg Publications, 2007

[17] Fujikake, M., Kojima, O., Fukushima, S. - Analysis of fabric tension structures, Comput. Struct. 32, 537-547, 1989

[18] Moncrieff, E., Topping, B. H. V. - Computer methods for the generation of membrane cutting patterns. Comput. Struct. 37, 441-450 1990.

[19] Scheck, H.-J. - The force density method for form finding and computation of general networks. In Computer Methods in Applied Mechanics and Engineering, Vol. 3, pages 115–134, 1974

[20] Pauletti, R. M. O., Pimenta, P. M. – The Natural Force Density Method for Shape Finding of Taut Structures, Comput. Methods Appl. Mech. Engrg. 197, 4419–4428, 2008

[21] Barnes, M.R. - Form-finding and analysis of tension space structures by dynamic relaxation: Ph.D. thesis, City University, London, 1977

[22] Maurin, B., Motro, R. - The surface stress density method as a form-finding tool for tensile membranes. In Engineering Structures, Vol. 20, pages 712–719, 1998

[23] Haftka R.T., Gürdal Z. - Elements of structural optimization. 3rd Edition, Kluwer Academic Publishers, 1996

[24] Kirsh U. - Structural optimization, fundamentals and applications, Springer-Verlag, 1993

[25] Sindel, F., Nouri-Baranger, T., Trompette, P. - Including optimization for the conception of fabric structures. Journal of Computers and Structures, 79 (26-28), 2451–2459, 2001

[26] Vanderplaats, G.N. - Numerical optimization techniques for engineering design with applications. Mc Graw-Hill, 1984

[27] Vanderplaats, G.N. - Multidisciplinary design optimization. Vanderplaats Research & Developement, Inc., 2007

[28] Rao, S.S. - Engineering Optimization - Theory and Practice, 4th Edn., John Wiley & Sons, Inc., 2009

[29] Philipp, B., K.U. Bletzinger – Hybrid Structures – Enlarging the Design Space of Architectural Membranes, Journal of International Association for Shell and Spatial Structures, Vol. 54, No.4, December n.178, 2013

[30] E. Ramm, K.U. Bletzinger, R. Reitinger - Shape optimization of shell structures, IASS Bull. 34, 103‑121, 1993

[31] K.U. Bletzinger, E. Ramm - Structural optimization and form finding of light weight structures, Comput. Struct. 79, 2053‑ 2062, 2001

[32] K.U. Bletzinger, E. Ramm - A general finite element approach to the form finding of tensile structures by the updated reference strategy, Int. J. Space Struct. 14, 131‑146, 1999

[33] K.U. Bletzinger, M. Firl, J. Linhard, R. Wüchner - Optimal shapes of mechanically motivated surfaces. In Computer Methods in Applied Mechanics and Engineering, 2008

[34] K.U. Bletzinger, R. Wüchner, F. Daoud, N. Camprubi - Computational methods for form finding and optimization of shells

and membranes. In ComputerMethods in Applied Mechanics and Engineering, Vol. 194, pages 3438–3452, 2005

[35] K.U. Bletzinger – Structural Optimization, Lecture Notes, TU Munich, 2014

[36] K.U. Bletzinger – Numerical Theory, Lecture Notes, IMS Institute, Dessau 2016

[37] Bendsøe, M., Sigmund, O. - Topology Optimization, Springer Verlag, 2003

[38] Nocedal J., Wright, S. J. – Numerical Optimization, Springer Verlag, 1999

[39] Arora, J.S. - Introduction to Optimum Design. Elsevier Academic Press, 2004

[40] Christensen, P.W., Klarbing, A. – An Introduction to Structural Optimization, Springer Verlag, 2009

[41] Martínez, D., Velho, L., Carvalho, P. C., - Computing Geodesics on Triangular Meshes, Conputer and Graphics, 2005

[42] Yang Q S, Tan F, Wang X F. Loading and wrinkling analysis of membrane structures. Sci China Tech Sci, 54: 2597-2604, 2011

[43] Chen, W., Wei, P., Bao, Y. – Surface Flattening based on Linear-Elastic Finite Element Method, World Academy of Science, Engineering and Technology 55, 2011

[44] Lewis, W.J. - Tension Structures − Form and Behaviour, Thomas Telford Ltd, London, 2003

[45] K. Ishii, Membrane Structures in Japan, SPS Publ. Co., Tokyo, 1995