Bauhaus University Weimar

Faculty of Civil Engineering

Chair of Modelling and Simulation of Structures

Structural Optimization of Composite Cross-Sections and Elements using Energy Methods

by

Igor Kavrakov

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

September, 2014

Declaration

The author declares that this document is his own original work, with the exception where explicit reference is made to the work of other authors. It has not been submitted for any other qualification to this or any other university.

Igor Kavrakov

Abstract

Structural optimization has gained considerable attention in the design of structural engineering structures, especially in the preliminary phase.

This study introduces an unconventional approach for structural optimization by utilizing the Energy method with Integral Material Behavior (EIM), based on the Lagrange's principle of minimum potential energy. An automated two-level optimization search process is proposed, which integrates the EIM, as an alternative method for nonlinear structural analysis, and the bilevel optimization. The proposed procedure secures the equilibrium through minimizing the potential energy on one level, and on a higher level, a design objective function. For this, the most robust strategy of bilevel optimization, the nested method is used. The function of the potential energy is investigated along with its instabilities for physical nonlinear analysis through principle examples, by which the advantages and limitations using this method are reviewed. Furthermore, optimization algorithms are discussed.

A numerical MATLAB based fully functional code is developed for nonlinear cross section, element and 2D frame analysis, utilizing different finite elements and is verified against existing EIM programs. As a proof of concept, the method is applied on selected examples using this code on cross section and element level. For the former one a comparison is made with standard procedure, by employing the equilibrium equations within the constrains. The validation of the element level was proven by a theoretical solution of an arch bridge and finally, a truss bridge is optimized. Most of the principle examples are chosen to be adequate for the everyday engineering practice, to demonstrate the effectiveness of the proposed method.

This study implies that with further development, this method could become just as competitive as the conventional structural optimization techniques using the Finite Element Method.

Acknowledgments

I would like to express my deepest gratitude to Prof. Guido Morgenthal, for his immense support and encouragement throughout my studies. His profound, yet simple approach of solving sophisticated engineering problems was truly inspirational and never seizes to amaze me.

Furthermore, I am entirely grateful for the time Dr. Hans-Georg Timmler has devoted for my endless questions and doubts. Without his vision and ideas, especially for the practical application, none of this would be possible.

I would also like to thank Peter Olney for shearing his invaluable experience with the EIM method and particularly the numerical implementation.

I also want to thank the whole department of Modeling and Simulation of Structures for their constant technical support and insightful comments.

Particular thanks goes to my friends, whom I consider family, for their constant support.

I am gratefull to the Deutcher Akademischer Austausch Dienst (DAAD) and the Bauhaus University - Weimar for granting me with scholarships to finish this course.

Last and most important, I would like to thank my family, whom I own everything, for their constant unconditional support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Structural form has always been a challenge for any great civilization. From the pyramids in Egypt, designed to sustain enormous self weight to the Roman arches which present ingenious solution of tall stable structure crossing, for that time, unimaginable spans sustained solely by compression, up to date, where longer, taller, slender and more economical structures are build. Historically, the engineers depended only on their intuition and engineering sense. In the modern era, the optimum design is just few clicks away, yet the engineering practice and intuition are never to be underestimated, since without them, an optimal design is just a lucky guess. Structural optimization has a considerable attention mostly in mechanical engineering, however with the modern technologies for light-weight materials and futuristic designs, it is getting its recognition in the structural engineering department rapidly. The increasing interest in this field is a direct result of rapid developments in structural analysis, optimization methods, moreover the availability of sophisticated, but inexpensive computers. Obtaining the optimal shape, by minimizing the weight, stresses, strains leads to more efficient, economical and futuristic design.

Generally, the structural optimization could be referred from three aspects, a structural one, which involves a method for structural analysis, an optimization aspect involving a mathematical procedure of minimization of a function and an intuitive one. These three aspects combined may lead to a successful solution, such as weight, volume or cost reduction. Any deficiencies in the optimization procedure, involving improper choice of optimization algorithm, inaccurate definition of the physical model or poor choice of initial conditions, within these three aspects, may very well lead to no or a unsatisfactory result, as it is proven commonly. Typical structural optimization procedures are developed using the Finite Element Method (FEM). Extensive research has been conducted for these procedures with many different approaches based on the equilibrium conditions as a constrain.

Within this study, an attempt for structural optimization is made, using the alternative method for nonlinear analysis, the Energy method with Integral description of Material behavior (EIM) based on the Lagrange principle of minimum potential energy. The idea is to couple the afore mentioned method, which itself involves minimization of the energy function as convex optimization problem, with the existing structural optimization procedures. Principally, on cross section level this is achievable by imposing the equilibrium conditions within the constrains of the optimization; however the challenge arises on element level when the distribution of internal forces occurs. The initial proposal was to minimize the two functions, the potential energy and one containing design parameters, simultaneously by employing the multi-objective optimization, which proved not to satisfy the equilibrium. Instead, a bilevel optimization was introduced.

The work is organized in seven chapters as follows. Chapter 2 reviews the background of the mechanical formulation of the EIM on cross section and element level, along with the variational principles on continuum level. Here, the ground rules and basic assumptions are established and serves as a foundation for the numerical code which was introduced further. In the end of this chapter, a comparison between the FEM and EIM is made.

In Chapter 3 the optimization techniques are presented as the second aspect of structural optimization. First the theoretical basis of convex optimization is reviewed along with its analytical solution, then an overview of algorithms used within the scope of this work along with recommendations for their usage is presented. Finally, a classification of the basic terms of structural optimization and their definitions are detailed.

A numerical MATLAB based code is introduced in Chapter 4, along with the numerical formulation of the EIM on cross section and element level. Different aspects of the code are presented along with recommendations for its usage. The idea is to formulate a modular code which would find its way in further research. In the second part of the chapter, the structural optimization with the EIM is discussed by a principle problem and the bilevel optimization is introduced.

First the verification with existing software, and then the implementation in selected structural examples on cross section and element level is demonstrated in Chapter 5 in order to demonstrate the capability and usability of the code.

Chapter 6 offers a summary of the work presented, along with conclusions and recommendations for further work.

Chapter 2

Formulation of the EIM

In this chapter, the theoretical principles used for formulation of the calculation model are outlined. The energy methods as variational principle for solution of continuous-system mathematical models and the Energy method with Integral Material behavior (EIM) ground rules for the numerical implementation are established and compared to the standard Finite Element Method (FEM). The relations are adopted from [9] [45] [47] [36] , which serve as a reference for detailed mathematical relations. Here only the fundamental principles of the EIM will be reviewed .
The analysis of an engineering systems requires an idealization of the system into a form that can be solved, the formulation of the mathematical model, the solution of this model and the interpretation of the results. The formulation of the mathematical model describing engineering problems is divided in two categories, discrete and continuum-mechanics-based models. Within the discrete systems, or known as lumped-parameters systems, the formulation of the governing equations is directly described by the solution of a set of algebraic equations with a finite number of state variables. In continuous systems, the formulation of the system response is govern by differential equations. The exact solution of the continuous systems is possible for relatively simple mathematical models, and therefore numerical procedures must be employed in order to reduce the contentious systems to a discrete idealization. There are two approaches of formulation of the continuous-system [9]:

- Differential formulation
- Variational formulation

In the differential formulation the system is defined with three coupled differential equations in terms of state variable, formulating the Navier differential equation [20]. This equation couples the equilibrium conditions, constitutive law and kinematic relations with respect to boundary conditions which brings the structural analysis to boundary value problem. As an alternative to the direct method, the state of equilibrium can be also obtained using an extremum, a variational formulation. An extremum problem consists of locating a set of values (state variables) x_i, $i = 1...n$, for which a given functional $\Pi(x_i...x_n)$ is a maximum, minimum or has a saddle point with respect to the state variables. The condition for obtaining the equation for state variables is $\delta\Pi = 0$. The variational formulation is usually reffed as the energy form since it is based on the formulation of the energy of the system. The reasons of using the variational formulation are discussed in the cited sources. From the previous observation it can be concluded that the variational formulation can be closely correlated to mathematical optimization problem and can be directly utilized to derive appropriate optimization formulation [47].

2.1 Energy principles

Energy is, in general, defined as a measure of the amount of work done on a body or by it. The reference point for the energy can be chosen arbitrarily, since only differences of energies play a role in physical processes. Adding a constant energy does not change a system's physical behavior. There are many forms of energy like kinetic, potential, magnetic, radiant, nuclear, gravitational, thermal, heat etc. Here, it is dealt with the mechanical energy which is the sum of the potential and kinetic energy.

The energy law of mechanics can be deduced from the mathematical interpretation of the Gauss integral sentence, the so-called 1st. Green functional for two independent functions, the displacements $\mathbf{u}$[1] and $\boldsymbol{\sigma}$ the stresses [20] [36]:

$$\int\limits_V (A\mathbf{u})^T \boldsymbol{\sigma}\,\mathrm{d}V + \int\limits_V \mathbf{u}^T A^T \boldsymbol{\sigma}\,\mathrm{d}V - \int\limits_S \mathbf{u}^T A_S^T \boldsymbol{\sigma}\,\mathrm{d}S = 0. \tag{2.1}$$

Equation (2.1) states that the energy of a force state (which is an equilibrated group) performed on a compatibly deformed displacement state disappears. The total energy consists of internal and external part presented by the first two terms and boundary surface energy parts presented by the last term. V represents the volume over which the energy is integrated, and S the surface of the volume where static and kinematic boundary conditions are applied. A and A_S are differential operators containing differentiation rules, where the latter one is used for the static boundary conditions.
The equilibrium and the kinematic equation are given in equation (2.1). These and the constitutive law which, is discussed later, are the 3 fundamental equations. The equilibrium equation (2.2a) connects the internal and external forces through the static differential operator D_k and the kinematic equation (2.2b) connects the internal and external displacements through the kinematic differential operator D_e. With the assumption of geometrically linear behavior D_k and D_e are adjoint differential operators:

$$D_k\mathbf{u} - \boldsymbol{\epsilon} = 0, \qquad \in V, \tag{2.2a}$$

$$D_e\boldsymbol{\sigma} - \mathbf{f}_0 = 0, \qquad \in V, \tag{2.2b}$$

$$A = D_k = D_e^T. \tag{2.2c}$$

With the division of the boundary surface S, two parts can be distinguished: the part S_u where the external displacements $\mathbf{u_{s,0}}$ are defined and the part S_f where the external

[1] all vectors, the stress and strain tensor are embolden throughout this work; in case of discretization, they are capitalized. Matrices are capitalized in italics, and scalars are in italics

reaction forces are acting $\mathbf{f_{s,0}}$. The first part is defined on equation (2.3b) as kinematic and the second as static boundary conditions on equation (2.3c):

$$S = S_u + S_f, \tag{2.3a}$$

$$\mathbf{u} - \mathbf{u_{s,0}} = 0, \qquad \in S_u, \tag{2.3b}$$

$$A_s^T \boldsymbol{\sigma} - \mathbf{f_{s,0}} = 0, \quad \in S_f. \tag{2.3c}$$

By inputting the previous relations in equation (2.1) an formulation for a volume V for statically admissible stresses and geometrically permissible displacements, taking into account the equilibrium, compatibility and boundary conditions [36]:

$$\int\limits_V (D_k\mathbf{u})^T \boldsymbol{\sigma}\,\mathrm{d}V + \int\limits_V \mathbf{u}^T D_e \boldsymbol{\sigma}\,\mathrm{d}V - \int\limits_{S_u} \mathbf{u_{s,0}}^T A_S^T \boldsymbol{\sigma}\,\mathrm{d}S - \int\limits_{S_f} \mathbf{u}^T \mathbf{f_{s,0}} \boldsymbol{\sigma}\,\mathrm{d}S = 0. \tag{2.4}$$

There are 2 classic variational principles using the energy law of mechanics:

- Principle of virtual displacements (virtual work)
- Principle of virtual forces (conjugate virtual work)

The first one uses the displacements as state variables and it is known as the *kinematic formulation* of mechanical problems, and in the latter one, known as the *static formulation*, stresses are used as state variables. There are also mixed formulations which will not be discussed in this case.

As previously argued, the variational formulation can be derived by invoking the stationary of the potential $\delta\Pi = 0$, by applying virtual forces (stresses) $\delta\boldsymbol{\sigma}$, defined in on equation (2.5a), or virtual displacements $\delta\mathbf{u}$ as shown in (2.5b). In both cases, the term "virtual δ" refers to an infinite small value:

$$\boldsymbol{\sigma} = \overline{\boldsymbol{\sigma}} + \delta\boldsymbol{\sigma}, \tag{2.5a}$$

$$\mathbf{u} = \overline{\mathbf{u}} + \delta\mathbf{u}. \tag{2.5b}$$

Here it is dealt with the kinematic formulation, therefore only the principle of virtual displacements will be derived which is defined as [20]:

"*A force state ($\mathbf{f_0}$,$\mathbf{f_{s,0}}$) is in (at least weak) equilibrium if we can find an arbitrary virtual, but kinematically compatible displacement state ($\delta\mathbf{u}$,$\delta\mathbf{u_{s,0}}$) for which the sum of virtual work disappears ($\delta\Pi = 0$)*"

Therefore, applying the equation (2.5b) in (2.4), for which the energy caused by existing displacements $\overline{\mathbf{u}}$ must satisfy $\overline{\Pi} = 0$ after (2.4), and applying homogeneous boundary conditions for the virtual displacements:

$$\delta\mathbf{u} = 0, \quad \delta\mathbf{u} \in S_u, \tag{2.6}$$

the virtual content must satisfy, which yields to the variational formulation after the Lagrange principle:

$$\delta\Pi = \int\limits_V (D_k\delta\mathbf{u})^T \boldsymbol{\sigma}\,\mathrm{d}V + \int\limits_V \delta\mathbf{u}^T D_e \boldsymbol{\sigma}\,\mathrm{d}V - \int\limits_{S_f} \delta\mathbf{u}^T \mathbf{f_{s,0}} \boldsymbol{\sigma}\,\mathrm{d}S = 0. \tag{2.7}$$

Applying the static and kinematic requirements (2.1):

$$\delta\Pi = \int\limits_V \delta\boldsymbol{\epsilon}\boldsymbol{\sigma}\,\mathrm{d}V + \int\limits_V \delta\mathbf{u}^T\mathbf{f_0}\,\mathrm{d}V - \int\limits_{S_f} \delta\mathbf{u}^T\mathbf{f_{s,0}}\boldsymbol{\sigma}\,\mathrm{d}S = 0. \tag{2.8}$$

The total strains could be decomposed in three parts, the pre-strains $\boldsymbol{\epsilon}_0$ related to initial conditions, the elastic $\boldsymbol{\epsilon}_{el}$ and the plastic $\boldsymbol{\epsilon}_{pl}$ part. The material behavior in the elastic range could be described by the stress-strain relation (2.9b). The plastic strains must fulfill the *associated flow rule* [45] given in (2.9c) related to the yield function Φ:

$$\boldsymbol{\epsilon} = \boldsymbol{\epsilon}_0 + \boldsymbol{\epsilon}_{el} + \boldsymbol{\epsilon}_{pl}, \tag{2.9a}$$

$$\boldsymbol{\sigma} = f(\boldsymbol{\epsilon}_{el}), \tag{2.9b}$$

$$\boldsymbol{\epsilon}_{pl} = \boldsymbol{\Phi}^T\lambda. \tag{2.9c}$$

The plastic multiplier λ always needs to be positive for the yielding part $\lambda \geq 0$. Substituting the previous relations in the variational formulation:

$$\delta\Pi = \int\limits_V \delta\boldsymbol{\epsilon}_{el}\boldsymbol{\sigma}\,\mathrm{d}V + \int\limits_V \delta\boldsymbol{\epsilon}_{pl}\boldsymbol{\sigma}\,\mathrm{d}V + \int\limits_V \delta\mathbf{u}^T\mathbf{f_0}\,\mathrm{d}V - \int\limits_{S_f} \delta\mathbf{u}^T\mathbf{f_{s,0}}\boldsymbol{\sigma}\,\mathrm{d}S = 0. \tag{2.10}$$

The plasticity condition is given by (2.11). With this condition the variation of the plastic strains vanishes when the strains are in the elastic region. This relation and (2.9) concludes the constitutive law which is the last of the 3 fundamental equations:

$$\boldsymbol{\Phi}\boldsymbol{\sigma} = S_0. \tag{2.11}$$

Employing the constitutive law, the variational formulation can be written in a form:

$$\delta\Pi = \int\limits_V \delta\boldsymbol{\epsilon}_{el}\boldsymbol{\sigma}\,\mathrm{d}V + \int\limits_V \delta\lambda^T S_0\,\mathrm{d}V + \int\limits_V \delta\mathbf{u}^T\mathbf{f_0}\,\mathrm{d}V - \int\limits_{S_f} \delta\mathbf{u}^T\mathbf{f_{s,0}}\boldsymbol{\sigma}\,\mathrm{d}S = 0. \tag{2.12}$$

From this form the principle of minimum of the total potential energy is derived in [24]:
”*Of all kinematically admissible displacements a body can have, the one is the actual which minimizes the total potential energy of the body*”

$$\Pi = \int\limits_V F(\boldsymbol{\epsilon}_{el})\boldsymbol{\sigma}\,\mathrm{d}V + \int\limits_V \lambda^T S_0\,\mathrm{d}V + \int\limits_V \mathbf{u}^T\mathbf{f_0}\,\mathrm{d}V - \int\limits_{S_f} \mathbf{u}^T\mathbf{f_{s,0}}\boldsymbol{\sigma}\,\mathrm{d}S \to min. \tag{2.13}$$

This last expression is analog to the equilibrium conditions and satisfied constrains. The formulation has a unique solution only for stable material. In this formulation, unlike the static one, the differential operators D_k and D_e are not subjected to restrictive conditions; therefore geometrically nonlinear behavior also could be considered.

2.2 Nonlinear analysis with EIM

The EIM is based on two basic principles: the *extremum* principle by Lagrange, the minimum of the potential energy which was derived in the previous section and on the *integral description of the material behavior*:

$$\Pi = \Pi_e + \Pi_i \rightarrow min. \tag{2.14}$$

where, Π_e denotes the external potential energy due to loads and Π_i the internal potential energy (strain energy). With discretisation of the continuum, with the help of weak formulation, on elements and cross section and integrating the strain energy, described with the integral description, by using standard nonlinear optimization solvers computationally fast and effective geometrically and physically nonlinear structural problems can be solved. This method was introduced by Raue in [30],[31],[35],[34],[16] and [33].
Before outlining the basis, certain assumptions which are made should be defined:

- The Bernoulli hypothesis applies, stating that the cross sections normal to the axis of the element remain plane during the deformation process.
- Strain energy from torsion and shear deflections is neglected.
- Loads are acting in the shear center of the cross section.
- The strains between different materials in a cross sections at the interface are the same, i.e there is rigid bond between them.
- For pre-strains, if implemented, the Iljushin's criterion is not violated, i.e prestrains due to different loading can be superimposed.

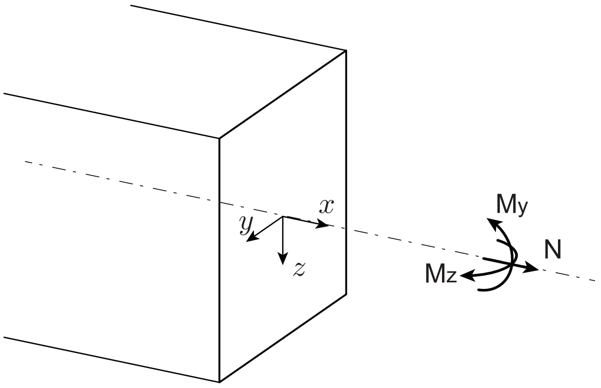


Figure 2.1: External loads on cross section.

The integral description of the material is the basis of the EIM. The constitutive law is the first step of the description:

$$\sigma = \sigma(\epsilon). \tag{2.15}$$

The integral description is obtained by integrating this law over the strains resulting with the specific strain energy W, the F and the Φ function, which describe the same behavior of one specific material. The latter two functions are used within the strain integration over complex geometries, in order the internal potential energy Π_i to be obtained:

$$W = W(\epsilon) = \int_0^\epsilon \sigma(\epsilon)\,\mathrm{d}\epsilon, \tag{2.16a}$$

$$F = F(\epsilon) = \int_0^\epsilon F(\epsilon)\,\mathrm{d}\epsilon, \tag{2.16b}$$

$$\Phi = \Phi(\epsilon) = \int_0^\epsilon \Phi(\epsilon)\,\mathrm{d}\epsilon. \tag{2.16c}$$

Arbitrary materials can be considered if the deformation behavior can be described in terms of stress-strain relations, with different or the same function for compression or tension laws, irregardless if it is linear or nonlinear function. As an example, for a typical linear elastic material defined by the Hooke's law, with Young's modulus E, the relations with schematic plots are given below.

$$\sigma(\epsilon) = E\epsilon \qquad W(\epsilon) = \frac{E}{2}\epsilon^2 \qquad F(\epsilon) = \frac{E}{6}\epsilon^3 \qquad \Phi(\epsilon) = \frac{E}{24}\epsilon^4 \tag{2.17}$$

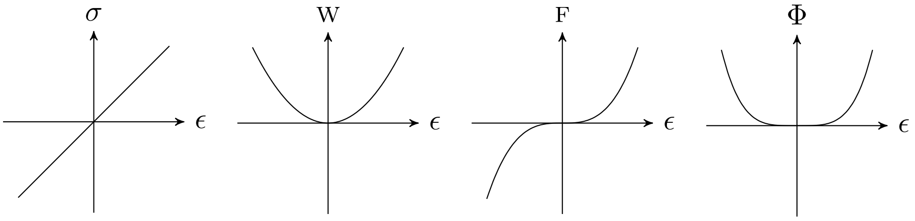


Figure 2.2: Example of σ, W, F and Φ for linear elastic material.

2.2.1 Cross section formulation

Comparability of deformations

As previously mentioned in the assumptions, the Bernoulli hypothesis applies for this method; therefore, with the remaining of the normal cross sections to the element axes plane during deformation, the strain $\boldsymbol{\epsilon} = \boldsymbol{\epsilon}(y, z)$ at arbitrary point in the cross-section with coordinates y and z, could be described by linear function:

$$\epsilon_x(y, z) = \epsilon(y, z) = \epsilon_0 + \kappa_y y + \kappa_z z. \tag{2.18}$$

In the latter equation, x is the a coordinate along a vector normal to the plane of the cross section, ϵ_0 defines the strain at the origin of the coordinate system , κ_y and κ_z are the curvatures (or gradients) of the stress plane with respect to the y and z axis respectively:

$$\nabla\epsilon = \begin{bmatrix} \dfrac{\partial\epsilon}{\partial y} \\ \dfrac{\partial\epsilon}{\partial z} \end{bmatrix} = \begin{bmatrix} \kappa_y \\ \kappa_z \end{bmatrix}. \tag{2.19}$$

In case of biaxial bending, a new coordinate system exists with coordinates η and ζ, at which along the η axis the strain is constant as shown on Figure 2.3. The relation between the two coordinate systems is defined with the angle ϕ with the following equation:

$$\cos\phi = \frac{\kappa_z}{\kappa}, \qquad \sin\phi = \frac{\kappa_y}{\kappa}, \tag{2.20a}$$

$$\eta = y\cos\phi + z\sin\phi = \frac{\kappa_z}{\kappa}y - \frac{\kappa_y}{\kappa}z, \tag{2.20b}$$

$$\zeta = -y\sin\phi + z\cos\phi = \frac{\kappa_y}{\kappa}y + \frac{\kappa_z}{\kappa}z, \qquad (\kappa \neq 0). \tag{2.20c}$$

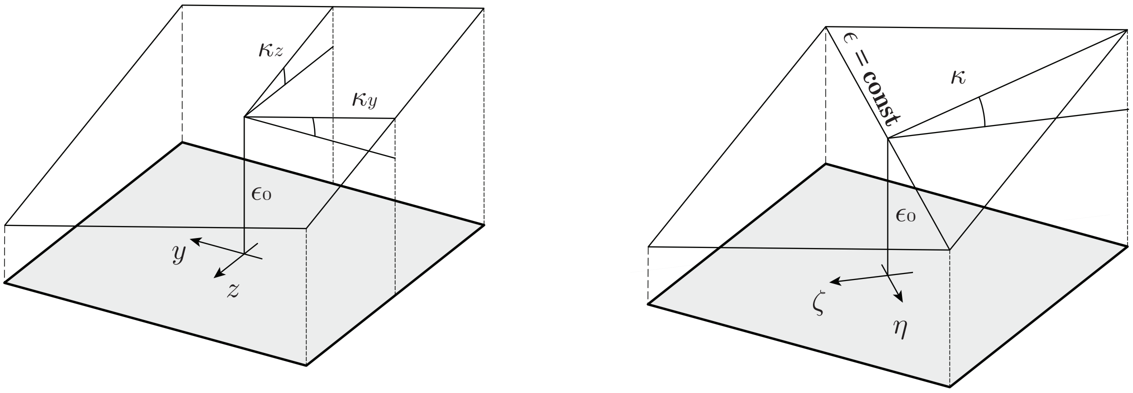


Figure 2.3: Coordinate systems for strain-state by biaxial bending.

The relations applies only if the norm of the gradient κ of the strain field is not equal to 0. The norm of the gradient is defined according to:

$$\kappa = \sqrt{\kappa_y^2 + \kappa_z^2}. \tag{2.21}$$

With this transformation the strain at any point of the cross section could be described only with respect to the ζ axes . If the norm of the gradient $\kappa = 0$, there is constant strains along the cross-section:

$$\epsilon = \begin{cases} \epsilon_0 + \kappa\zeta, & (\kappa \neq 0), \\ \epsilon_0, & (\kappa = 0). \end{cases} \tag{2.22}$$

If the angle $\phi = 0$, then it is a uni-axial bending case, and the coordinates y, z are identical to η and ζ respectively.

Total potential energy

The strain energy Π_i^C of a cross section with area A can be obtained by integrating the specific strain energy over the area $W(y,z)$ defined in equation (2.16a):

$$\Pi_i^C = \iint\limits_A W(y,z)\,\mathrm{d}y\,\mathrm{d}z = \iint\limits_A W[\boldsymbol{\epsilon}(y,z)]\,\mathrm{d}y\,\mathrm{d}z. \tag{2.23}$$

Raue in [30] made the ansatz that this formulation of the energy could be transformed into a line integral by Gauss's theorem. Then integrated over the perimeter S, would result in the total internal potential energy of the cross section.

Taking into account that the functions defined in equation (2.16) are 2-dimensional scalar fields and that the gradient of a scalar field is a vector field, the gradient of $\Phi[\boldsymbol{\epsilon}(y,z)]$ is defined by the 2-dimensional vector field $\mathbf{v}$:

$$\mathbf{v} = \begin{bmatrix} v_y(y,z) \\ v_z(y,z) \end{bmatrix} = \nabla\Phi = \begin{bmatrix} \dfrac{\partial\Phi}{\partial y} \\ \dfrac{\partial\Phi}{\partial z} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial\Phi}{\partial\epsilon}\dfrac{\partial\epsilon}{\partial y} \\ \dfrac{\partial\Phi}{\partial\epsilon}\dfrac{\partial\epsilon}{\partial z} \end{bmatrix}. \tag{2.24}$$

With this relation, the curl of the $\mathbf{v}$ is defined by:

$$\nabla\times\mathbf{v} = \frac{\partial v_z}{\partial y} - \frac{\partial v_y}{\partial z} = \kappa_z\frac{\partial F}{\partial y} - \kappa_y\frac{\partial F}{\partial z} = \kappa_z\frac{\mathrm{d}F}{\mathrm{d}\epsilon}\frac{\partial\epsilon}{\partial y} - \kappa_y\frac{\mathrm{d}F}{\mathrm{d}\epsilon}\frac{\partial\epsilon}{\partial z} = (\kappa_z\kappa_y - \kappa_y\kappa_z)\frac{\mathrm{d}F}{\mathrm{d}\epsilon} = 0. \tag{2.25}$$

With the $\nabla\times\mathbf{v} = 0$, it is ensured that the $\mathbf{v}$ is a conservative field which, in physical systems, represent that the energy is conserved and the function $\Phi(y,z)$ is the potential. This being proven, $\mathbf{v}$ has the property that the line integral is path independent which will be calculated later.

The divergence of $\mathbf{v}$ results in a 2-dimensional scalar field:

$$\nabla\cdot\mathbf{v} = \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z} = \kappa_y\frac{\partial F}{\partial y} + \kappa_z\frac{\partial F}{\partial z} = \kappa_y^2\frac{\mathrm{d}F}{\mathrm{d}\epsilon} + \kappa_z^2\frac{\mathrm{d}F}{\mathrm{d}\epsilon} = \kappa^2 W(y,z), \tag{2.26}$$

and if equation (2.25) is taken into account and applied:

$$\nabla\cdot(\nabla\Phi) = \Delta\Phi(y,z) = \kappa^2 W(y,z). \tag{2.27}$$

The internal strain energy, with the substitution of equation (2.26) into (2.23), could be expressed as:

$$\Pi_i^C = \iint\limits_A W(y,z)\,\mathrm{d}y\,\mathrm{d}z = \frac{1}{\kappa^2}\iint\limits_A \left[\frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z}\right]\mathrm{d}y\,\mathrm{d}z = \frac{1}{\kappa^2}\iint\limits_A \left[\kappa_y\frac{\partial F}{\partial y} + \kappa_z\frac{\partial F}{\partial z}\right]\mathrm{d}y\,\mathrm{d}z. \tag{2.28}$$

Comparing this with the Gauss's theorem:

$$\iint\limits_A \left(\frac{\partial Z}{\partial y} - \frac{\partial Y}{\partial z}\right)\mathrm{d}y\,\mathrm{d}z = \oint\limits_S (Y\,\mathrm{d}y + Z\,\mathrm{d}z). \tag{2.29}$$

yields to the final formulation of the strain energy of the cross section, described by a integral along the contour (perimeter) S and taking into account the deformation coordinate system transformations:

$$\Pi_i^C = \frac{1}{\kappa^2} \iint\limits_A \left[\kappa_y \frac{\partial F}{\partial y} + \kappa_z \frac{\partial F}{\partial z} \right] \mathrm{d}y \, \mathrm{d}z = \oint\limits_S (-\frac{\kappa_z}{\kappa^2} F \, \mathrm{d}y + \frac{\kappa_y}{\kappa^2} F \, \mathrm{d}z) = -\frac{1}{\kappa} \oint\limits_S F \, \mathrm{d}\eta. \tag{2.30}$$

For composite cross section composed of n partitions or *subcross sections*, as later referred, instead of an integral, the strain energy is a sum:

$$\Pi_i^C = \Pi_i^C(\epsilon_0, \kappa_y, \kappa_z) = \sum_{i=1}^{n} \Pi_{i,i}^C. \tag{2.31}$$

For a cross section externally loaded by an axial force N and/or two moments M_y and M_z around the z and y axes respectively, the external potential energy Π_e^C is defined by:

$$\Pi_e^C = \Pi_e^C(\epsilon_0, \kappa_y, \kappa_z) = -(N\epsilon_0 + M_y\kappa_z + M_z\kappa_y). \tag{2.32}$$

It should be noted that the strain and external potential energy are functions dependent on three parameters ϵ_0, κ_y and κ_z. With the external and internal energy being calculated, the Lagrange principle presented in (2.14) could be applied and by finding its minimum i.e. a stationary point in the energy function, the equilibrium on cross section level is established. Since the potential energy is a convex function[2], the gradient must be equal to 0:

$$\nabla \Pi^C(\epsilon_0^*, \kappa_y^*, \kappa_z^*) = 0. \tag{2.33}$$

The values ϵ_0^*, κ_y^*, κ_z^* are the values of the three parameters at the minimum of the potential energy. The gradient is defined by the three partial derivatives:

$$\frac{\partial \Pi^C}{\partial \epsilon_0} = \frac{\partial \Pi_i^C}{\partial \epsilon_0} - N = 0, \tag{2.34a}$$

$$\frac{\partial \Pi^C}{\partial \kappa_y} = \frac{\partial \Pi_i^C}{\partial \kappa_y} - M_z = 0, \tag{2.34b}$$

$$\frac{\partial \Pi^C}{\partial \kappa_z} = \frac{\partial \Pi_i^C}{\partial \kappa_z} - M_y = 0. \tag{2.34c}$$

The latter equations describe the equilibrium conditions on a cross section level.

2.2.2 Element formulation

Compliance of the compatibility conditions

The cross section formulation can be extended on a element level with the correlation between the displacements and deformations. The unknown parameters for the cross section minimization of the energy, ϵ_0, κ_y and κ_z depend on, now the unknowns for the element level, the displacements $u(x)$, $v(x)$ and $w(x)$. With the assumption of

[2] Valid only for some instances, described further chapters.

geometrically linear behavior the relation between displacements and deformations can be defined as:

$$\epsilon_0 = u'(x), \tag{2.35a}$$

$$\kappa_z = -w''(x), \tag{2.35b}$$

$$\kappa_y = -v''(x). \tag{2.35c}$$

For geometrically nonlinear line-like elements, taking into account the theory of moderate rotations (w' or $v' < 10°$) which neglects the second order derivative of the displacements within the Taylor's series expansion of the deformations, the longitudinal strains are expressed by:

$$\epsilon_0 = u'(x) + \frac{1}{2}(w')^2 + \frac{1}{2}(v')^2. \tag{2.36}$$

The relations between the curvatures remain the same. There are further relations, assuming different deflected shapes, however they are beyond the scope of the subject mater. Further relations are elaborated in [36].

Total potential energy

Considering the element on Figure 2.4 with length l, the strain energy of the element Π_i^E is obtained by integrating the strain energy of the cross section Π_i^C over the length l.

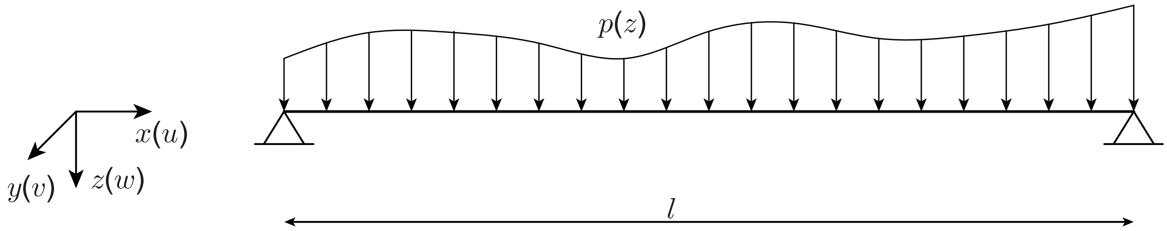


Figure 2.4: Element formulation.

$$\Pi_i^E = \int_0^l \Pi_i^C \, \mathrm{d}x. \tag{2.37}$$

The external potential energy Π_e^E is obtained by integrating the product of the external loads and the corresponding displacements over the length l:

$$\Pi_e^E = -\int_0^l \left[p_x(x)u(x) + p_y(x)v(x) + p_z(x)w(x) + m_y(x)\varphi_y(x) + m_z(x)\varphi_z(x)\right] \mathrm{d}x. \tag{2.38}$$

Total potential energy on a element level Π^E could be obtained by simple superposition of the strain Π_i^E and external energy Π_e^E and its minimization the equilibrium conditions are met. In this case, as previously mentioned, the displacements are the unknown parameters.

2.3 Comparison of EIM and classical FEM

The principle used in the classical FEM is derived from the virtual displacement method which is outlined on the beginning in this chapter. The derivation could be found in [9]. For nonlinear analysis with the FEA a stiffness degradation has to be employed. The most famous is the Newton-Raphson which is an iterative procedure of computing the stiffness matrix. Figure 2.5 present the procedure compared between the two methods for displacement formulation. In the classical FEM, the residual between the externally applied forces $\mathbf{P}$[3] and forces which correspond to the stresses $\mathbf{F}$ is set to zero and the resulting nonlinear set of equations is solved iteratively, e.g. with a Newton-Raphson scheme. The quadratic convergence is the major advantage of the Newton-Raphson method and under the condition that the initial point $\mathbf{U}_0$ considerably close to the solution the convergence is fast. The assumption of this initial point varies from problem to problem and it could be considered as the major disadvantage of this method. The solutions obtained with these two methods differ in some aspects: The FEM finds all equilibrium configurations, including stable and non-stable ones, but the starting point has to be close enough to a solution, otherwise there might be no convergence. In contrast, EIM finds only stable equilibrium, but a solution is always found when the potential is convex[4], independent of the quality of the starting point.

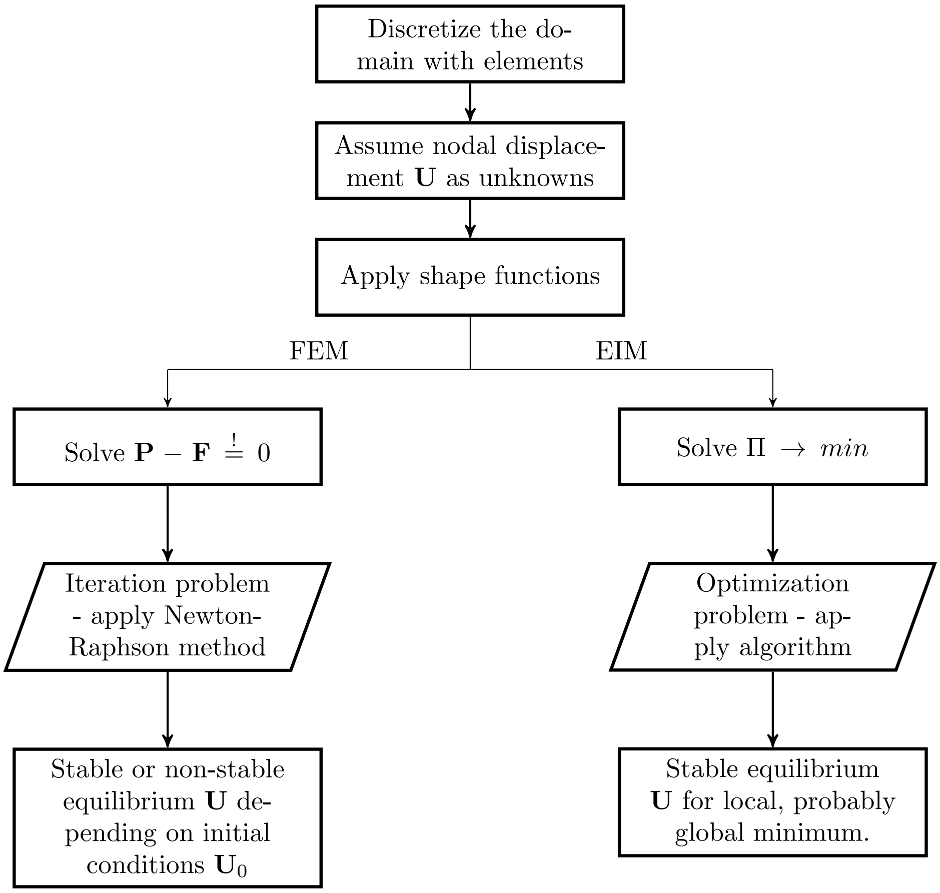

Discretize the domain with elements
Assume nodal displacement $\mathbf{U}$ as unknowns
Apply shape functions
FEM
EIM
Solve $\mathbf{P} - \mathbf{F} \overset{!}{=} 0$
Solve $\Pi \rightarrow min$
Iteration problem - apply Newton-Raphson method
Optimization problem - apply algorithm
Stable or non-stable equilibrium $\mathbf{U}$ depending on initial conditions $\mathbf{U}_0$
Stable equilibrium $\mathbf{U}$ for local, probably global minimum.

Figure 2.5: Flowchart of the EIM and FEM.

[3] The embolden capitalized vectors for the physical parameters refer to discretized systems.

[4] The convexity of the energy is discussed within the further chapters.

Chapter 3

Optimization

In engineering, many tasks require optimization, whether for calibrating numerical according to experimental results or structural optimization of shape and parameters. Here, first the most common mathematical optimization algorithms will be reviewed for single and multi-objective problem and in the end an overview of structural optimization will be given. The optimization techniques within this work are used on two levels: minimizing the potential energy for the EIM and for structural optimization. In general the subject mater is a very vast area and most of the principles are adopted from [21][18][37][19][47][44].

3.1 Basic principles and classification

The first task approaching an optimization problem is to find the mathematical formulation of it. Although this looks like a basic procedure, usually it is the main challenge in optimization problems. There are three main considerations that are of major importance before approaching an optimization problem:

- **Goal** - What is the purpose of the optimization or which is the final outcome that is desirable to be achieved? This usually represent a function, or a set of functions that should be minimized or maximized refereed as *objective function(s).*
- **Variables** - Which parameters could be subjected to a change in order to attain the goal? One has to be also acquainted whether the variables are representing physical or only numerical property of the problem.
- **Bounds** - Which are the explicit restriction of the variables? If the variable is for example a physical property as area, then the obvious bound would be that the area could not be negative.
- **Constrains** - In contrast to bounds which directly puts limits on the optimization space, constraints are functions of the variables.

Once these considerations are established, it could be moved on with the choosing of the algorithm and then finally solving the problem. The standard form to formulate a constrained optimization problem (OP) is as follows [18]:

$$OP : \begin{cases} \min\limits_{\mathbf{x}} f(\mathbf{x}), \\ \text{subject to} \begin{cases} g_j(\mathbf{x}) \leq 0, & j = 1 \ldots m_g, \\ h_k(\mathbf{x}) = 0, & k = 1 \ldots m_h, \\ x_i^l \leq x_i \leq x_i^u, & i = 1 \ldots n. \end{cases} \end{cases} \tag{3.1}$$

or, the mathematical formulation:

$$\min_{\mathbf{x} \in \mathbf{X}} f(\mathbf{x}), \quad \text{with} \quad \mathbf{X} = \{\mathbb{R}^n | g_j(\mathbf{x}) \leq 0; h_k(\mathbf{x}) = 0; x_i^l \leq x_i \leq x_i^u\}. \tag{3.2}$$

This could be interpreted in other words as: to find a minimum or maximum of an *objective function* $f(\mathbf{x})$ or a set of s functions $f(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_s(\mathbf{x})]^T$ depended n *parameters* $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$which can be taken from the *feasible set* $\mathbf{X}$ i.e $\mathbf{x} \in \mathbf{X}$. $\mathbf{X}$ is a subspace of $\mathbb{R}^n$ i.e $\mathbf{X} \subseteq \mathbb{R}^n$ in which the *bounds* $x_i^{lower} \leq x_i \leq x_i^{upper}$, the *equality constrains* $h_k(\mathbf{x}) = 0$ and *inequality constrains* $g_k(\mathbf{x}) \leq 0$ are satisfied. Generally optimization problems could be classified in the following categories [21]:

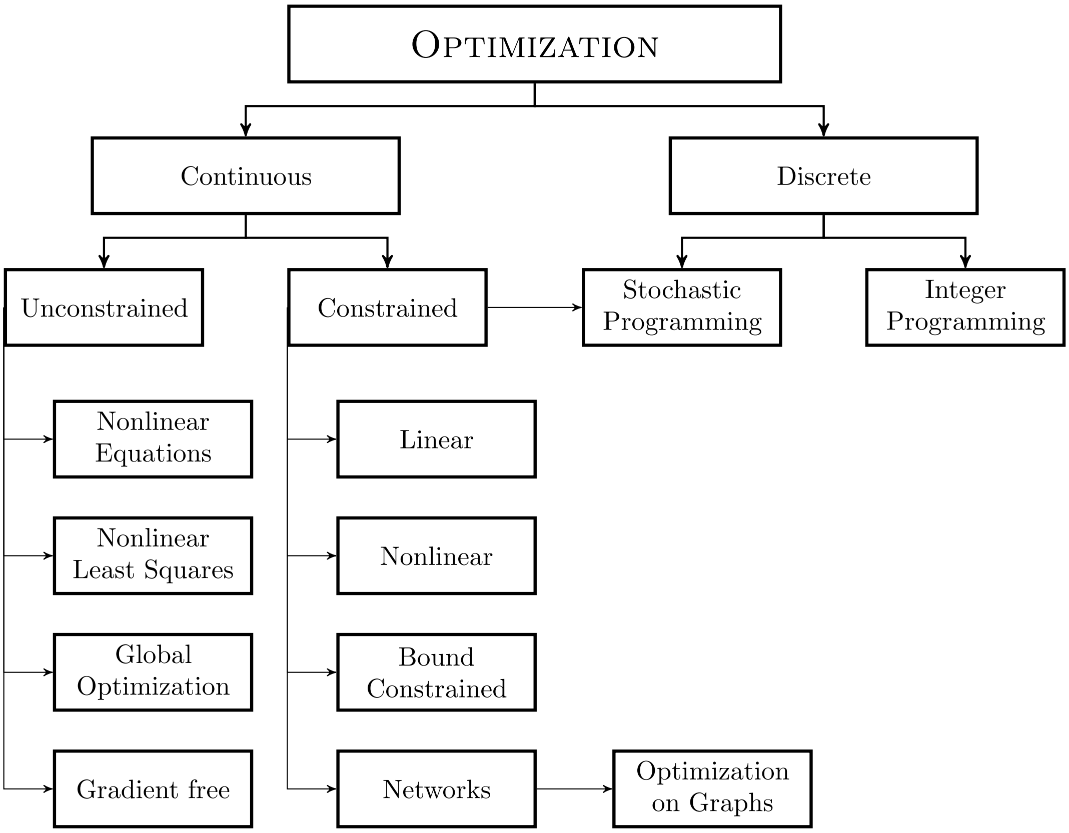


Figure 3.1: Optimization tree adopted from Lahmer [21].

Dealing with more than one objective function the optimization is referred as *multi-objective optimization*. The feasible set $\mathbf{X}$ can be finite or infinite. The former case is denoted as *unconstrained optimization* and in the latter as *constrained* one. Categorizing the dependence of the objective function on the parameter as linear or nonlinear it can also be distinguished between *linear* and *nonlinear* optimization. With respect to the definition of the next step of the optimization, if it is defined by some "path" it is referred as *deterministic* or continuous optimization, or on the contrary if it is random with a certain probability distribution and each computation it takes different path it is called *stochastic* optimization. Furthermore, the minimum found with the optimization can be a local or a global minimum which categorizes the optimization as a *local* or a *global* one.

3.2 Optimization algorithms

For a set of relatively simple nonnonlinear equations, there is an analytical way of computing the local minimum; however the analytical way of solving an optimization problem is not always a straightforward task. Computing the Hessian for sophisticated functions for real optimization issues is usually impossible and therefore numerical algorithms must be implemented. Here, an introduction in the methods used in the scope of this work is given.

3.2.1 Unconstrained nonlinear optimization

If there are no constrains or bounds on the optimization problem, equation (3.1) yields to an unconstrained optimization problem:

$$OP : \min_{\mathbf{x}} f(\mathbf{x}). \tag{3.3}$$

The local minimum $\mathbf{x}^*$, referred as *stationary point*, of an unconstrained optimization problem with objective function $f(\mathbf{x})$, can be defined within two, so called, optimality conditions [21]:

- **Necessary condition**
  Let f be twice differentiable. The necessary condition for $\mathbf{x} = \mathbf{x}^*$ is the gradient $\nabla f(\mathbf{x}^*)$ to be 0 (3.4a) and the Hessian $\nabla^2 f(\mathbf{x}^*)$ is positive semidefinite (3.4c). The Hessian matrix is symmetric and is the Jacobian of ∇f, shown in equation (3.4b).

$$\nabla f(\mathbf{x}^*) = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n}\right]^T = 0, \tag{3.4a}$$

$$\nabla^2 f(\mathbf{x}^*)_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}, \tag{3.4b}$$

$$(\mathbf{x}^*)^T \nabla^2 f(\mathbf{x}^*) \mathbf{x}^* \geq 0. \tag{3.4c}$$

- **Sufficient condition**
  Let f be twice differentiable in a neighborhood of $\mathbf{x}^*$. If the gradient ∇f is equal to 0 and the Hessian $\nabla^2 f(\mathbf{x}^*)$ is positive definite, $\mathbf{x}^*$ is a local minimum of f:

$$\nabla f(\mathbf{x}^*) = 0, \tag{3.5a}$$

$$(\mathbf{x}^*)^T \nabla^2 f(\mathbf{x}^*) \mathbf{x}^* > 0. \tag{3.5b}$$

Equations (3.4) to (3.5) define the analytical solution of an optimization problem. Usually for complex optimization problems the minimum is computed numerically with optimization algorithms. Choosing an appropriate algorithm always depends on the objective function and initial conditions, especially if the optimization problem is not linear. In case there is some *a priori* knowledge for the optimization problem, using deterministic algorithms usually leads to satisfactory results. Dealing with smooth functions, the Newton and quasi-Newton methods have proven to be very efficient and in the other case where the objective function is not differentiable, usually the direct search methods such as the Neadler-Mead simplex algorithm are used due to the low computational cost. However it is not always possible to identify the analytical form of the objective function. In this case, it is dealt with so called black-box optimization where probabilistic methods are dominant. These methods are popular within the state of the art optimization strategies and are attractive in the field of research. Numerous algorithms such as the Particle Swarm Optimization (PSO), the Simulated Annealing (SA), Evolutionary Algorithms (EA) and the Monte-Carlo method were developed in the recent history. From these, an introduction of the EA group will be made, due its use within this study. For further information with respect to the probabilistic algorithms, please refer to [39], [46] and [21]. There exists also the hybrid methods, where after a global minimum is found using a probabilistic method, a deterministic one is applied in order to prove the stationary of the minimum. As a conclusion, the *No Free Lunch Theorem* by Wolpert & Macready always applies when dealing with optimization problems which states that any two optimization algorithms are equivalent when their performance is averaged across all possible problems [48], [23].

Newton methods

Newton methods are based on minimizing successively a quadratic approximation of the objective function with numerical computation of the gradient and Hessian at each step in order to prove the optimality conditions. The quadratic approximation of the objective function is obtained using the Taylor's series until the second order:

$$\begin{aligned} f(\mathbf{x}) \approx q_k(\mathbf{x}) = {} & f(\mathbf{x}^{k-1}) + \nabla f(\mathbf{x}^{k-1})^T (\mathbf{x}^k - \mathbf{x}^{k-1}) \\ & + \frac{1}{2}(\mathbf{x}^k - \mathbf{x}^{k-1}) \nabla^2 f(\mathbf{x}^{k-1})(\mathbf{x}^k - \mathbf{x}^{k-1}). \end{aligned} \tag{3.6}$$

A stationary point is obtained if the gradient $\nabla f = 0$:

$$q_k(\mathbf{x}) \to \min \;\text{ if }\; \nabla q_k(\mathbf{x}) = 0. \tag{3.7}$$

Then, the updated solution for $\mathbf{x}^{k+1}$ is:

$$\nabla q_k(\mathbf{x}) = \nabla f(\mathbf{x}^k) + \nabla^2 f(\mathbf{x}^k)(\mathbf{x}^{k+1} - \mathbf{x}^k), \tag{3.8a}$$

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \nabla^2 f(\mathbf{x}^k)^{-1} \nabla f(\mathbf{x}^k). \tag{3.8b}$$

This is only valid under the assumption of positive definite Hessian. Normally in numerical procedures the Hessian matrix is not inverted, instead the increment $(\mathbf{x}^{k+1} - \mathbf{x}^k)$ is found by solving a set of nonlinear equations and the solution vector $\mathbf{d}_k$ is then updated using this increment. The algorithm, as described in [21] is the following:

Algorithm 1 Local Newton method

```
Set k=0
Choose $\mathbf{x}^0 \in \mathbb{R}^n, \epsilon > 0$
while $||\nabla f(\mathbf{x}^k)|| > \epsilon$ do
    $\mathbf{d}^k$ solves $\nabla^2 f(\mathbf{x}^k)\mathbf{d}^k = -\nabla f(\mathbf{x}^k)$
    $\mathbf{x}^{k+1} = \mathbf{x}^k + \mathbf{d}^k$
    $k = k + 1$
```

The algorithm is refereed as the local Newton method. The global method requires step-size control ether by the *Line Search* or *Trust-Region* methods. *Line Search* method uses a magnification factor for the solution vector $\mathbf{d}_k$, β which scales the step size, and within the *Trust-Region* method, $\mathbf{d}_k$ is constrained to a domain, where it is expected the resulting direction vector $\mathbf{d}_k$ could yield a satisfactory reduction. For the sake of brevity, these methods will not be derived within this work; however they are explained thoroughly in [18], [21] and [37]. Computation of the inverse of the Hessian is or solving the system of nonlinear equations for $\mathbf{d}^k$ is computationally costly procedure. Therefore, with an approximation of the Hessian H_k by an iterative procedure, the quasi-Newton methods were established. The general scheme is [18]:

1. Compute a search direction $\mathbf{d}^k = -H_k^{-1} \nabla^2 f(\mathbf{x}^k)$.
2. Find $\mathbf{x}^{k+1} = \mathbf{x}^k + \beta \mathbf{d}^k$.
3. Use $\mathbf{x}^k$, $\mathbf{x}^{k+1}$, $\nabla f(\mathbf{x}^k)$, $\nabla f(\mathbf{x}^{k+1})$ and H_k to compute H_{k+1}.

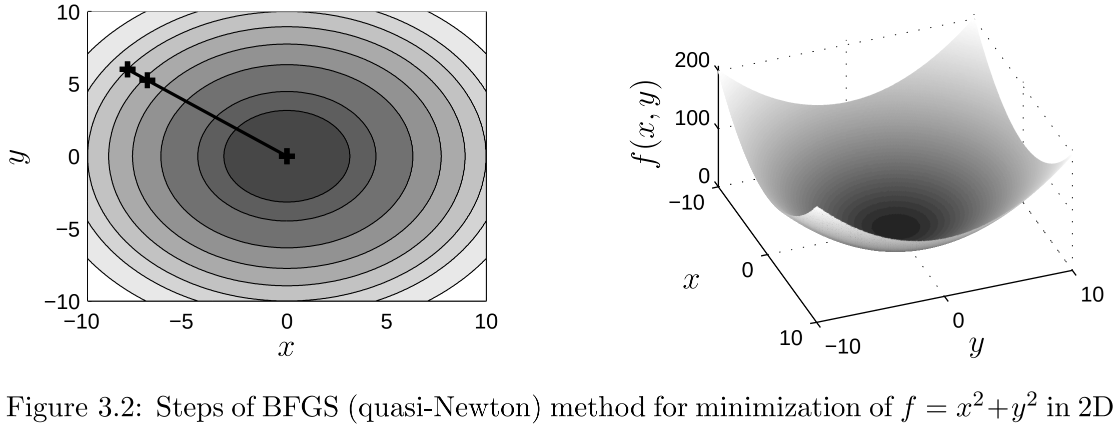


Figure 3.2: Steps of BFGS (quasi-Newton) method for minimization of $f = x^2 + y^2$ in 2D with initial conditions $(x_0; y_0) = (-8; 6)$ (left) and 3D representation of the function (right).

Some of the quasi-Newton methods used in the modern optimizers are: Broyden's method; Broyden-Fletcher-Goldfarb-Schano method (BFGS) and Davidon-Fletcher-Powell formula. On Figure 3.2 it is shown the minimization of sphere function with the BFGS method. It could be concluded that only few steps are required when the objective function is smooth and if the line search algorithm for step-size control is used.

Nelder-Mead simplex algorithm

Nelder and Mead [25] have developed a gradient free, deterministic algorithm based on a simplex which moves in the function space, adjusting its size with respect to the function value at the vertices. The simplex is one dimension higher then the function space $\mathbb{R}^n$ i.e it is $n+1$ dimensional simplex. In each iteration the worst vertex from the edges of the simplex is replaced with four basic moves and eventually the minimum will be surrounded from the simplex edges. The presentation of this algorithm will be after [18] and [21].

In this algorithm the vertices (or the vectors containing the parameters) are aligned with respect to the objective function values:

$$f(\mathbf{x}_1) \leq f(\mathbf{x}_2) \leq \cdots \leq f(\mathbf{x}_{n+1}). \tag{3.9}$$

After realizing the worst vertex $\mathbf{x}_{n+1}$ the idea is to replace it with a new point:

$$\mathbf{x}(\mu) = (1+\mu)\bar{\mathbf{x}} - \mu\mathbf{x}_{n+1}, \tag{3.10}$$

where the $\bar{\mathbf{x}}$ denotes the centroid of the simplex:

$$\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i. \tag{3.11}$$

As an example, On Figure 3.3 the moves of a 3 dimensional simplex in 2D function space are given. The vertex $\mathbf{x}_3$ gives the worst function value and the vertex $\mathbf{x}_1$ the best. On Figure 3.4 the minimization of the sphere function using the Nelder-Mead algorithm is presented. Here, only the lowest value of the simplex's vertices are recorded for clarity.

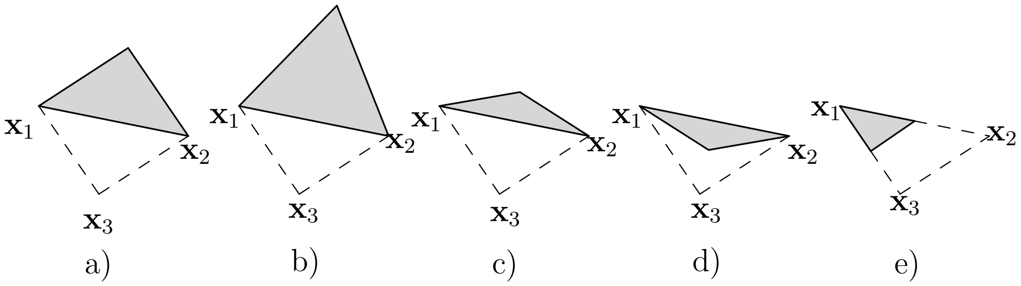


Figure 3.3: Moves of a 3 dimensional simplex in 2D function space: a) Reflect; b) Expand; c) Outside Contraction d) Inside Contraction e) Shrink.

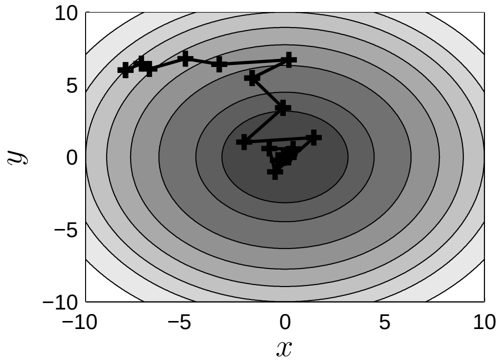


Figure 3.4: The lowest vertex of the simplex of the Nelder-Mead algorithm for minimization of $f = x^2 + y^2$ for each iteration.

The properties of this algorithm are advantageous when it is not dealt with a smooth objective function. One disadvantage is the no-guarantee of convergence since it is not proven mathematically i.e there could be stagnation at a non-optimal point; however in practice their performance is generally good [18], especially in the engineering practice since due to the a-priori knowledge of the optimization problem.

The four basic operations that the simplex can perform and converge within tolerance ϵ are formulated in the algorithm [21]:

Algorithm 2 Nelder-Mead Simplex Algorithm

```
Set values $-1 \leq \mu_{ic} \leq 0 \leq \mu_{oc} \leq \mu_r \leq \mu_e$; typically $-1 < -0.5 < 0 < 0.5 < 1 < 2$
Evaluate f at all $n+1$ vertices and sort according to (3.9)
Set $f_{count} = n + 1$
while $f(\mathbf{x}_{n+1}) - f(\mathbf{x}_1) > \epsilon$ do
    Compute $\bar{\mathbf{x}}$ (3.11), $\mathbf{x}(\mu_r)$ (3.10) and $f_r = f(\mathbf{x}(\mu_r))$. $f_{count} = f_{count} + 1$    ▷ (a)
    Reflect: If $f(\mathbf{x}_1) \leq f_r < f(\mathbf{x}_n)$, replace $\mathbf{x}_{n+1}$ with $\mathbf{x}(\mu_r)$ and go to (g)    ▷ (b)
    Expand: If $f_r < f(\mathbf{x}_1)$ compute $f_e = f(\mathbf{x}(\mu_e))$    ▷ (c)
    If $f_e < f_r$, replace $\mathbf{x}_{n+1}$ with $\mathbf{x}(\mu_e)$; otherwise replace $\mathbf{x}_{n+1}$ with $\mathbf{x}(\mu_r)$,
    $f_{count} = f_{count} + 1$, go to (g)
    Outside Contraction: If $f(\mathbf{x}_n) \leq f_r < f(\mathbf{x}_{n+1})$, compute $f_{oc} = f(\mathbf{x}(\mu_{oc}))$,    ▷ (d)
    $f_{count} = f_{count} + 1$. If $f_{oc} \leq f_r$ replace $\mathbf{x}_{n+1}$ with $\mathbf{x}(\mu_{oc})$
    and go to (b); otherwise to (f)
    Inside Contruction: If $f_r \geq f(\mathbf{x}_{n+1})$ compute $f_{ic} = f(\mathbf{x}(\mu_{ic}))$.    ▷ (e)
    $f_{count} = f_{count} + 1$. If $f_{ic} < f(\mathbf{x}_{n+1})$, replace $\mathbf{x}_{n+1}$ with $\mathbf{x}(\mu_{ic})$
    and go to (g); otherwise to (f)
    Shrink: If $f_{count} > k_{max} - n$, exit. For $2 \leq i \leq n+1$:    ▷ (f)
    set $\mathbf{x}_i = \mathbf{x}_1 - (\mathbf{x}_i - \mathbf{x}_1)/2$, compute $f(x_i)$
    Sort: Sort the vertices of simple according to (3.9)    ▷ (g)
```

Evolutionary Algorithms

The Evolutionary Algorithms (EA) presents a group of algorithms based on the natural selection from the biologic evolution theory. They are stochastic, global and gradient free algorithms. Processes like *selection*, *mutation*, *recombination* (or *crossover*) are implemented within the search of the optimum value. First, the objective function, in this case called *fitness* function, is computed within the first generation of individuals. The fittest individuals are selected, denoted as *parents*, from which based on normal distribution within the range of parameters, children are created with the mutation and crossover processes. The later process occurs when a new entity is formed from the information of two parents, and within the mutation, with a stochastic nature, a offspring is formed from a single parent as a input. A brief interpretation of the algorithm is presented [39]:

Algorithm 3 General Scheme of an Evolutionary Algorithm in pseudo-code

```
Initialize population with random candidate solutions
Evaluate each candidate
while stoping criteria is met do
    Selection of the parents based on their fitness
    Recombination or crossover of the pairs of parents to get children or
    Mutation of the parents
    Evaluate of the new candidates
    Selection of the individuals for the next generation
```

Typical representatives of the EA group are the Genetic Algorithm (GA) and Evolution Strategies (ES). Mostly, the various subcategories of the evolutionary algorithms differ only in technical details, however the general structure of the algorithm is maintained. EA are a fine alternative in black-box optimization problems and unlike the previous two algorithm, the optimization process is a global one. However, the computation effort is higher then the deterministic algorithms. On Figure 3.5 the change of the population during the minimization of the sphere function with GA is shown. The initial population was created using random set of number within the range [-10,10]. For each member of the population the fitness function has to be evaluated within each iteration. Compared to the gradient method, significantly larger amount evaluations were needed; however, as mentioned before, dealing with black-box optimization, often the shape of the optimization function is not clear and this is when the EA have proved their superiority.

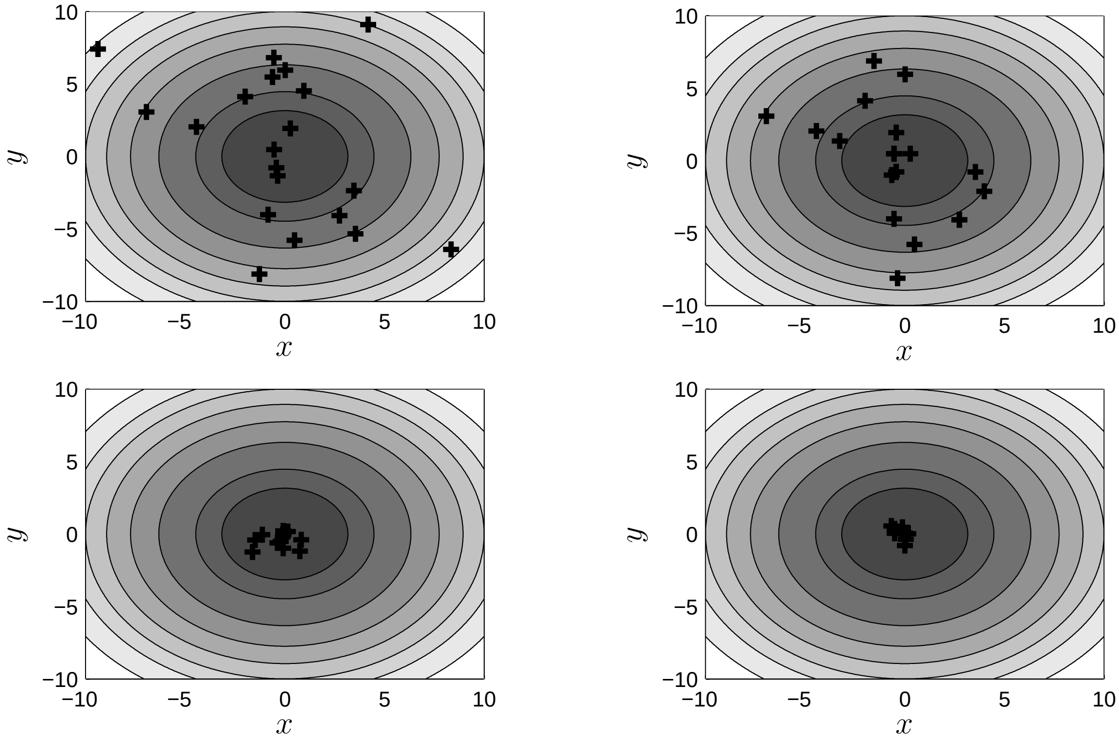


Figure 3.5: Population for minimization of $f = x^2 + y^2$ using genetic algorithm: from initial (top left) to final (bottom right).

3.2.2 Constrained nonlinear optimization

The approach of solving a constrained optimization, is somewhat different then the unconstrained one. In many methods, the unconstrained one is a sub-procedure of the constrained one. For unconstrained convex optimization problems, there is always a local which, is also a global minimum, if conditions defined in equations from (3.4) to (3.7) are satisfied. However, if the problem is not convex, there not always exists a minimum. In the case of constrained optimization problems, when the feasible set is compact, i.e. bounded and closed, a solution always exists (this is true for any continuous objective function, not necessarily convex). In the case where there exsists no convexity, a solution could not even be a stationary point, i.e. where $\nabla f(x) = 0$, but a point located on the boundary of the feasible set. This property is important to realize, and it will be used for the structural optimization using the EIM. If the objective function is strictly convex and the feasible set convex and compact, there exists exactly solution. The optimality conditions differ when there are constrains involved and were first presented by Kuhn and Tucker in [43], therefore they are also recognized as Karush-Kuhn-Tucker (KKT) optimality conditions. The optimization function defined in equation (3.1) with the introduction of the Lagrangian multiplier method and utilization of the optimization duality property $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^{m_g} \times \mathbb{R}^{m_h} \rightarrow \mathbb{R}$, derived in [44], yields to:

$$\mathcal{L}(\mathbf{x}, \lambda, \nu) = f(\mathbf{x}) + \sum_{j=1}^{m_g} \lambda_j g_j(\mathbf{x}) + \sum_{k=1}^{m_h} \nu_k h_k(\mathbf{x}). \tag{3.12}$$

The variables λ and ν are the so-called dual variables or Lagrangian multiplier vectors which are only positive when the constrain is active. Otherwise they are zero. If the problem is formulated in this way, at the solution point x^* for convex problems with the assumption of $\mathbf{g}$ being convex and differentiable and $\mathbf{h}$ affine and differentiable at $\mathbf{x}^*$, the KKT optimality conditions are *necessary* and *sufficient* conditions [44]:

$$\begin{aligned} \nabla f(\mathbf{x}) + \sum_{j=1}^{m_g} \lambda_j \nabla g_j(\mathbf{x}) + \sum_{k=1}^{m_h} \nu_k \nabla h_k(\mathbf{x}) = 0,& \\ g_j(\mathbf{x}^*) \leq 0, \quad j = 1 \ldots m_g,& \\ h_k(\mathbf{x}^*) = 0, \quad k = 1 \ldots m_h,& \\ \lambda_j^* \geq 0, \quad j = 1 \ldots m_g,& \\ \lambda_j^* g_j(\mathbf{x}^*) = 0, \quad j = 1 \ldots m_g.& \end{aligned} \tag{3.13}$$

If there is also a bound, beside the KKT conditions x^* should also satisfy $x_i^l \leq x_i^* \leq x_i^u$ for $i = 1 \ldots n$. For non-convex problems there are additional conditions which should be taken into consideration, however they will be not reviewed here. The closed form for the KKT conditions is available for only a handful of problems. The area of numerical constrained optimization is very vast, and generally there is a difficulty categorizing all of the algorithms. Methods for solving a constrained optimization problem in n variables and m constraints can be divided roughly into four categories that depend on the dimension of the space in which the accompanying algorithm works. Feasible direction methods (e.g. Zoutendijk method) work in $n - m$ space, penalty methods work in n space, dual and cutting plane methods work in m space, and Lagrangian methods (e.g. Sequential Linear Programming (SQP), Quadratic programing (QP)) work in $n + m$ space. Each of these approaches are founded on different aspects of Nonlinear Programming theory. Nevertheless, there are strong interconnections between them, both in the final form of implementation and in performance. The rates of convergence of most practical algorithms are determined by the structure of the Hessian of the Lagrangian, much like the structure of the Hessian of the objective function determines the rates of convergence for most unconstrained method [8]. Many of them are reviewed in [8], [44], [28]. In this case the penalty and barrier methods will be reviewed, according to [38], due to their explicit implementation within this work.

Penalty and barrier methods

Penalty and barrier method is a basic technique with the advantage of reformulation of the constrained optimization problem into a unconstrained one. Then simply algorithms for unconstrained optimization could be applied. If the basic objective function $f(\mathbf{x})$ is modified by a penalty function $\phi(\mathbf{w}, \mathbf{t})$, which penalizes any number $\mathbf{t}$ greater then 0 (from the point of view of minimization) with certain intensity defined by the weight $\mathbf{w}$, a new unconstrained optimization problem is formulated $\widetilde{f}(\mathbf{x})$.

The penalty function is defined $\phi(\mathbf{w},\mathbf{t})$ for $\mathbf{w} \geq 0$, $\mathbf{t} \in \mathbb{R}$ [38] if:

- ϕ is continuous.
- $\phi(\mathbf{w},\mathbf{t}) \geq 0$ for all $\mathbf{w}$ and $\mathbf{t}$.
- $\phi(\mathbf{w},\mathbf{t}) = 0.$ for $\mathbf{t} \leq 0$ and ϕ is strictly increasing for $\mathbf{w} > 0$ and $\mathbf{t} > 0$.

A typical example of a penalty function would be:

$$\phi(\mathbf{w},\mathbf{t}) = \begin{cases} 0, & \mathbf{t} < 0, \\ \mathbf{w}\mathbf{t}^n, & \mathbf{t} \geq 0. \end{cases} \tag{3.14}$$

In general case referring to standard optimization problem with equality and inequality constrains the $\widetilde{f}(\mathbf{x})$ would be:

$$\widetilde{f}(\mathbf{x}) = f(\mathbf{x}) + \sum_{j=1}^{m_g} \phi(w_{iq,j}, g_j(\mathbf{x})) + \sum_{k=1}^{m_h} (\phi(w_{eq,k}, h_k(\mathbf{x})) + \phi(w_{eq,k} - h_k(\mathbf{x}))). \tag{3.15}$$

where $\mathbf{w}_{eq}$ and $\mathbf{w}_{iq}$ are weights of the equality and inequality constrains that control how strongly the constrains will be enforced. With this modification, when the parameter enters values which are not in the feasible set $\mathbf{X}$, the penalty term increase the objective function which will result the algorithm to choose optimize within the feasible set. The advantage of the penalty method is the easy converting the constrained problem to an unconstrained one, and the constrains are "soft". However, it is greatly disadvantageous when the objective function is not defined outside the feasible.
To solve this problem,the barrier function methods are introduced, which are type of penalty functions where the objective function is always within the feasible set $\mathbf{X}$. These are generally only applicable for inequality constrains, i.e. only $g_k(\mathbf{x})$ are defined. The most common is the log barrier method which defines the modified objective function with:

$$\widetilde{f}(\mathbf{x}) = f(\mathbf{x}) + \sum_{j=1}^{m_g} w_{iq,j} ln(-g_j(\mathbf{x})). \tag{3.16}$$

However, even in the feasible region, the penalty term is non-zero, but it becomes "anti-penalty" if $g_j(\mathbf{x}) \leq -1$. The initial point for this method $\mathbf{x}_0$ should always be a feasible point, which in some instances it is not easily defined. In this case $\widetilde{f}(\mathbf{x})$ is only defined within the feasible region. On Figure 3.6, both methods are applied on a quadratic function $f(x) = x^2$ with inequality constrains $g(x) = 1 - x \leq 0$. A penalty function is applied of form (3.14) with $w = 6$, $t = 1 - x$, $n = 2$ with which the modified objective function $\widetilde{f_1}(x)$ was created and a barrier function with respect to (3.16) with $w = 1$ with which the second modified objective function was created $\widetilde{f_2}(x)$. The solutions $x_1^* = 0.85$ and $x_2^* = 1.3$ were obtained with the gradient method. It could be clearly seen that the penalty method violates the constrains, and the barrier one not. However they can be interpreted whether they are "good enough" or not, depending on the optimization problem and the tolerance and by modifying the weight of the function, to obtain a better solution.

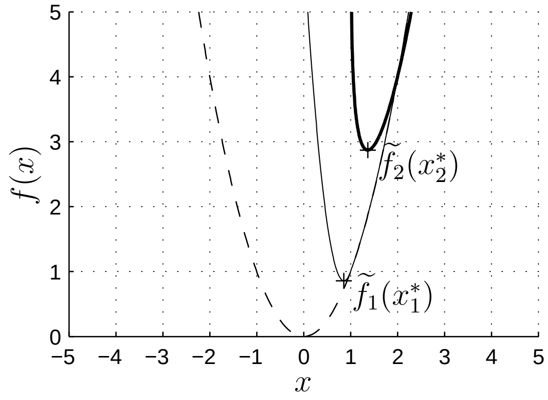


Figure 3.6: Modified objective functions and their minimum of $f(x) = x^2$(- -) using inequality constrains $g(x) = 1 - x \leq 0$, using penalty method $\widetilde{f}_1(x)$ (—) and barrier method $\widetilde{f}_2(x)$ (**—**).

3.2.3 Bilevel and multi-objective optimization

Although bilevel and multi-objective are optimization problems dealing with more than one objective function they are essentially different. Mathematical problems that consists an optimization problem in the constrains are bilevel or multilevel optimization problems, when they deal with more than one optimization problem in the constrains. On the other hand, multi-objective optimization concerns optimizing more than one objective functions simultaneously and usually have multiple solutions, whilst the bilevel has only one or none. Generally, it is not possible to use multi-objective algorithms for solving bilevel problems.

Bilevel optimization

A bilevel mathematical programming problem is formulated as [22]:

$$BOP : \begin{cases} \min\limits_{\mathbf{x},\mathbf{y}} F(\mathbf{x},\mathbf{y}), \\ \text{s.t} \quad G_l(\mathbf{x},\mathbf{y}) \leq 0, \qquad l = 1 \ldots m_l, \\ \text{where } \mathbf{x} \text{ solves} \begin{cases} \min\limits_{\mathbf{x}} f(\mathbf{x},\mathbf{y}), \\ \text{s.t} \quad g_j(\mathbf{x},\mathbf{y}) \leq 0, \quad j = 1 \ldots m_j. \end{cases} \end{cases} \tag{3.17}$$

where, $\mathbf{x} \in \mathbb{R}^n$; $\mathbf{y} \in \mathbb{R}^m$; $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$; $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$; $G : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^l$; $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^j$. The function $F(\mathbf{x},\mathbf{y})$ is generally defined as the upper level function with upper level variable set, and the function $f(\mathbf{x},\mathbf{y})$ as the lower level function with lower level variable set. When the lower function is unconstrained and convex, the problem is reffed as Mathematical Programs with Equilibrium Constrains (MPEC) [3].

This could be reformulated using the necessary condition of optimal solution, the gradient to be zero as:

$$BOP: \begin{cases} \min\limits_{\mathbf{x},\mathbf{y}} F(\mathbf{x},\mathbf{y}), \\ \text{s.t} \begin{cases} G_l(\mathbf{x},\mathbf{y}) \leq 0, & l = 1 \dots m_l, \\ \nabla_x f(\mathbf{x},\mathbf{y}) = 0. \end{cases} \end{cases} \tag{3.18}$$

The upper objective function is dependent on the outcome of the second one and is typically not a convex function. The two levels can be cooperating or conflicting depending on the objective functions. Methods for solution of bilevel optimization problems and MPEC are reviewed in [14] which include the vertex enumeration methods, penalty function method, branch-and-bound methods, the cutting plane algorithm etc. These are reviewed in the cited literature [12] [7] and will not be discussed here. Another common approach which will be included within this scope of work is the so called *Nested Optimization* method, where for every x lower level problem is solved completely. This method is advantageous since the lower level optimization is numerically solved completely independently of the upper one, i.e. the margin of numerical errors computing derivatives is less, however it is of high computational cost. Figure 3.7 presents a simple bilevel optimization problem, where the lower objective function is represented only at sequences $y^{(a)}$, $y^{(b)}$ and $y^{(c)}$. Generally this is a continuous function and x, y present sets of on more then one variable i.e $\mathbf{x} = [x_1, x_2, \dots, x_n]$ and $\mathbf{y} = [y_1, y_2, \dots, y_m]$.

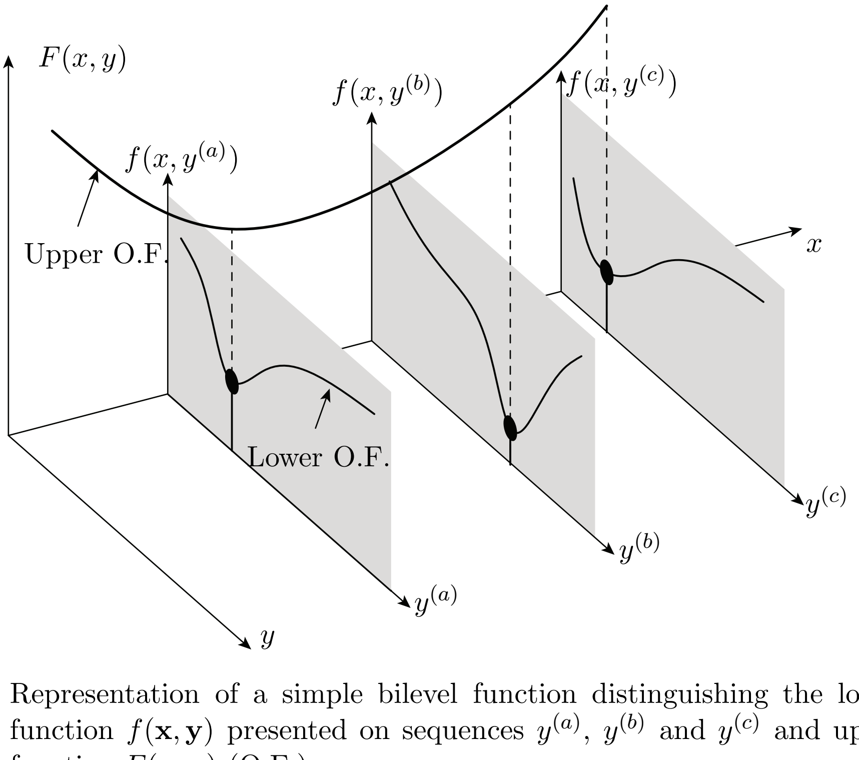


Figure 3.7: Representation of a simple bilevel function distinguishing the lower objective function $f(\mathbf{x},\mathbf{y})$ presented on sequences $y^{(a)}$, $y^{(b)}$ and $y^{(c)}$ and upper objective function $F(\mathbf{x},\mathbf{y})$ (O.F.).

Multi-objective optimization

In contrast to single objective optimization, a solution to a multi-objective problem is more of a concept than a definition. Typically, there is no single global solution, and it is often necessary to determine a set of points that all fit a predetermined definition for an optimum [5]. The predominant concept in defining an optimal point is that of *Pareto optimality* which is different than the term optimality defined for the single objective. The definition of an optimum is influenced by the decision maker's opinion and his preferences, whether the preference is expressed before the optimization within the apriori methods or after within the postpriori methods. Recall equation (3.1) where $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_s(\mathbf{x})]^T$ and $\mathbf{X}$ is a the feasible set. A vector $\mathbf{x}^* \in \mathbf{X}$ is said to be Pareto optimal for a multi-objective problem if all other vectors $\mathbf{x} \in \mathbf{X}$ have a higher value for at least one of the objective functions f_i with $i = 1, \ldots, s$ or have the same value for all the objective functions. Therefore, there are two types of Pareto optimal solutions [13]:

- A point $\mathbf{x}^*$ is said to be a weak Pareto optimum or a *weak* efficient solution for the multi-objective problem if and only if there is no $\mathbf{x} \in \mathbf{X}$ such that $f_i(\mathbf{x}) \leq f_i(\mathbf{x}^*)$ for all $i = 1, \ldots, s$.
- A point $\mathbf{x}^*$ is said to be a weak Pareto optimum or a *strict* efficient solution for the multi-objective problem if and only if there is no $\mathbf{x} \in \mathbf{X}$ such that $f_i(\mathbf{x}) < f_i(\mathbf{x}^*)$ for all $i = 1, \ldots, s$, with at least one strict inequality.

The Pareto curve for two objective function shown on Figure 3.8 represents the set of efficient solutions. The shape of this curve presents the compromise or the so called *trade off* [5] between the solutions. Here, the preference of the decision maker take part, which could be done through different methods. In general optimization problems, this curve is hard to compute especially when there are more than two objective functions.

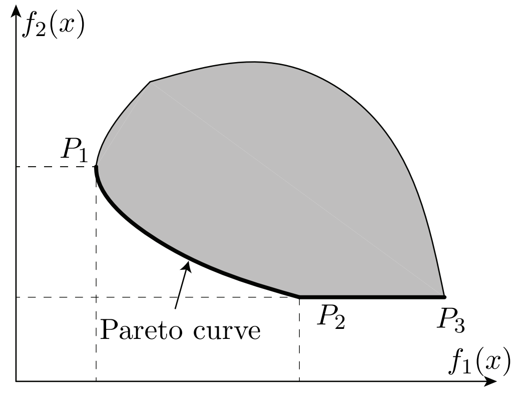


Figure 3.8: Pareto optimality for two objective functions. Point $P_1(f_1(x_1), f_2(x_1))$ is a strict solution; $P_2(f_1(x_2), f_2(x_2))$ and $P_3(f_1(x_3), f_2(x_3))$ are weak solutions.

Typically for methods with a priori articulation of preferences is to methods incorporate parameters, which are coefficients, exponents, constraint limits, etc. with which a single objective problem is formulated from the set of objective functions. However these parameters present additional degrees of freedom which alter the final solution and the

user could not obtain the desired solution. Most commonly used from these methods are the Weighted global criterion, Weighted sum, Weighted min-max, Weighted product,Bounded objective function, Lexicographic and the Goal programming method. In [5] they are well elaborated, although they are self-explanatory. Here it will be only dealt with the *Weighted sum method* since it is simple and satisfactorily results could be obtained with it and it is used within the scope of this work.
The Weighted sum method could be deducted from the Weighted global criterion method, with the parameter $p = 1$:

$$F(\mathbf{x}) = \sum_{i=1}^{s} \omega_i [f_i(\mathbf{x})]^p. \tag{3.19}$$

The ω is a vector of weights and typically $\sum_{i=1}^{s} \omega_i = 1$. Commonly the function which is of most importance is given the largest value (often the objective functions are normalized). There are techniques for their determination and the difficulties that could occur such as no guarantee of acceptable result despite the satisfactory assignment of weights.

3.3 Structural optimization

Understanding the algorithms for mathematical optimization given in the previous section are essential for the structural optimization. However in this case, the intuitive realization of the problem and initial design are of major importance just as well. Therefore, it could be concluded that structural optimization is iterative-intuitive process of making an assemblage of materials sustain loads in the best way and unlike the mathematical optimization, where the concept "as good as possible" apply, here the approach "good enough" is employed. This implies that the outcome should be evaluated from an engineering point of view, even in situations where that is not the mathematical optimum. Figure 3.9 presents the process of optimizing a mechanical structure.

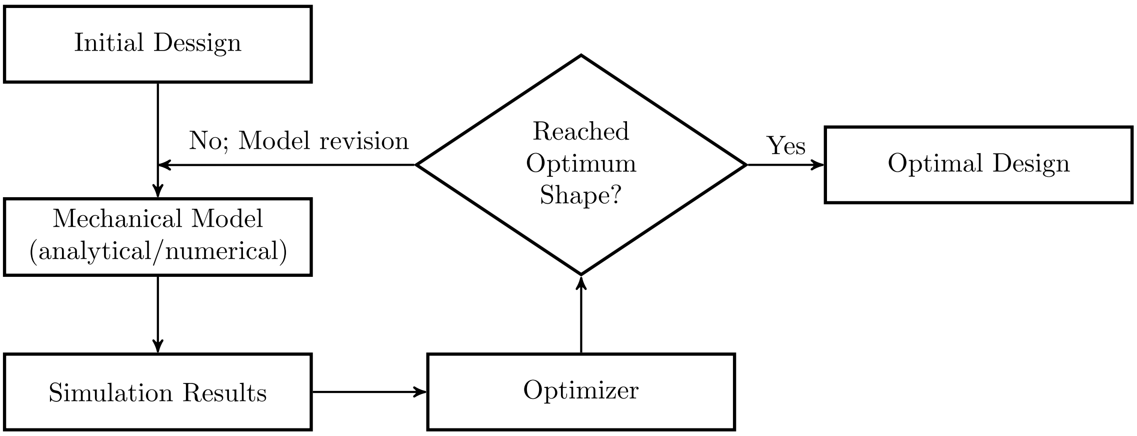


Figure 3.9: Optimization process of mechanical structures adopted from Lahmer [21].

The optimization problem given in equation (3.1) could be interpreted as a structural optimization problem (SO) [19]:

$$SO : \begin{cases} \min\limits_{\mathbf{x},\mathbf{y}} f(\mathbf{x},\mathbf{y}), \\ \text{subject to} \begin{cases} \text{behavioral constrains on } \mathbf{y}, \\ \text{design constrains on } \mathbf{x}, \\ \text{equilibrium constraint.} \end{cases} \end{cases} \tag{3.20}$$

In the case of SO, it is distinguished between two types of variables: *design variable* $\mathbf{x}$ and *state variable* $\mathbf{y}$. The design variable is a function or a vector that describes the design and which alter during optimization. It may represent choice of material or geometry, e.g. area of a bar or a shape. The state variable for a given design $\mathbf{x}$ is a function or a vector that represents the response of the mechanical model (e.g. displacement, stress or forces). The objective function f in this interpretation indicates the goodness of the design. Frequently used f measures weight, displacement in a given direction, effective stresses or strains etc. In the coming chapters it will be dealt with formulation of f thoroughly within examples. The behavioral constrains are constrains on y (e.g. displacement in a certain direction) and the design constrains are constrains on $\mathbf{x}$ (e.g. positive area). Obviously these two types of constrains could be combined and reformulated in mathematical form (3.1).
Typically, the equilibrium conditions are satisfied for a discrete problem using the FEM with the relation:

$$K\mathbf{U} = \mathbf{P} \tag{3.21}$$

where, K presents the stiffness matrix of the system, $\mathbf{U}$ the displacement vector and $\mathbf{P}$ the forcing vector. In the case of nonlinear analysis the stiffness matrix is calculated as described in Section 2.3. However, the equilibrium conditions could also be satisfied using the EIM method described in the previous chapter, with the Lagrange principle of potential energy in (2.14). This implies that for every optimization step, a stationary point must exist in order to obtain the correct state variables. The minimization of the energy is done by optimization techniques, usually by the gradient methods since it is a convex smooth function. Establishing this point, coupling structural optimization problem and the EIM method presents a bilevel optimization problem or precisely a MPEC problem reviewed in Section 3.2.3 with the total potential energy as a lower level objective function. Therefore a SO problem using the EIM method could be formulated in the following way:

$$SO : \begin{cases} \min\limits_{\mathbf{x},\mathbf{y}} f(\mathbf{x},\mathbf{y}), \\ \text{subject to} \begin{cases} \text{behavioral constrains on } \mathbf{y}, \\ \text{design constrains on } \mathbf{x}, \end{cases} \\ \text{where } \mathbf{y} \text{ solves} \begin{cases} \min\limits_{\mathbf{y}} \Pi(\mathbf{x},\mathbf{y}). \end{cases} \end{cases} \tag{3.22}$$

Since EIM is using the kinematic formulation, $\mathbf{y}$ are the displacements on an element level or deformations on a cross section level. It should be noted that the behavioral constrains could not be applied on the lower objective function, as a consequence of the constrained optimization property discussed in Section 3.2.2 i.e. in case there exists no convexity in the feasible set of $\mathbf{y}$, the solution of the lower objective function is not a stationary point and therefore there is no equilibrium. The application of the nested optimization and a discussion why multi-objective formulation could not be used in this case will be done in the next chapter.

The SO problems, if $\mathbf{x}$ represents a geometric feature of the structure, could be classified in three types of problems:

- *Sizing optimization*: Adoption of certain parameters of geometric objects e.g. $\mathbf{x}$ is cross sectional area, thickness or length of a truss bar.
- *Shape optimization*: Flexible changes and adoption of the boundaries of geometric objects, without formulation of any new boundaries. Here $\mathbf{x}$ represents the form or contour of the boundary of the structural domain e.g. outer form of a deck of a bridge.
- *Topology optimization*: The most general form of SO, where new boundaries are also formed i.e. the interior is allowed to change of a structure, a certain holes might appear or vanish, as an example, in discrete parameter systems bars removed from truss.

Ideally shape optimization is a subclass of topology optimization, but practical implementations are based on very different techniques. Concerning the relation between topology and sizing optimization, the situation is the opposite: from a fundamental point of view they are very different, but they are closely related from practical application [19].

Figure 3.10 present an example of sizing optimization, for discrete and continuous systems. In the first case, the area of the trusses is optimized, however it is constrained and it could not vanish i.e. those which do not contribute to the compliance have infinite small area.

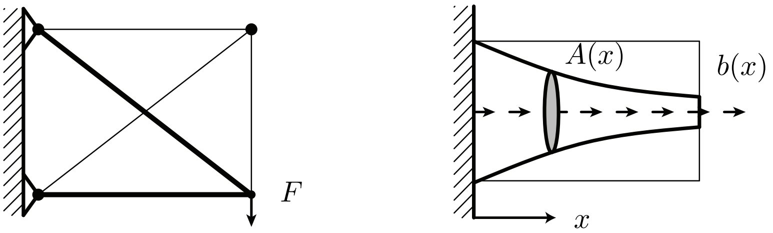


Figure 3.10: Sizing optimization: discrete (left) optimizing area of struts and continuous (right) optimizing of continuum subjected to body force $b(x)$.

Shape optimization is shown on Figure 3.11. Within the discrete case a boundary area is given for optimization of the position of the node. This depends of the forces acting on the node. The continuous case presents an aerodynamic shape optimization problem, where a bridge deck is optimized for wind action. The shape of the deck is a free with square as initial design. Numerically the shape is presented using splines or polynomials. Typical splines used in shape optimization are the Bézier splines, B-splines and the Nonuniform Rational B-Splines (NURBS). Within the scope of this work, mostly it will be dealt with discrete shape optimization, and in the few cases of one dimensional continuous, polynomials will be used, despite their high sensitivity. Therefore the splines will not be discussed, however they are thoroughly reviewed in [19].

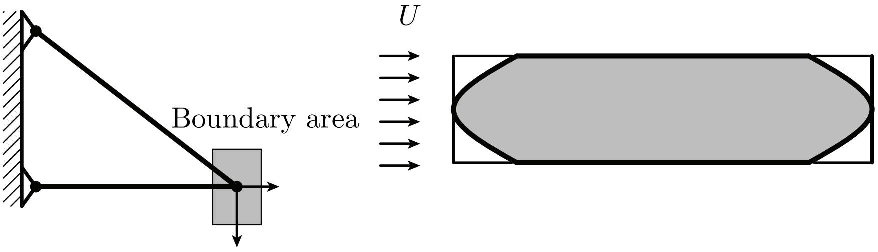


Figure 3.11: Shape optimization: discrete (left) optimizing the position of the node and continuous (right) optimizing a bridge deck subjected to aerodynamic forces.

Unlike the discrete sizing optimization, where the truss bars could not vanish, in topology the inactive bars can be removed. On Figure 3.12, a continuous topology optimization of simply supported beam. Typical methods using the FEM for this cause are the Evolutionary Structural Optimization (ESO) and Bidirectional Evolutionary Structural Optimization (BESO) which include removing of inactive finite elements from the mesh. However, these are not reviewed here.

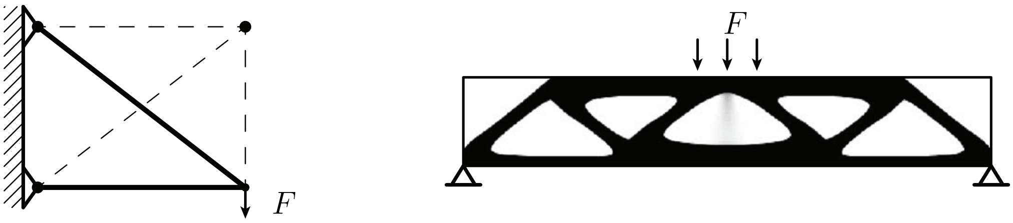


Figure 3.12: Topology optimization: discrete (left) optimizing the area of the struts which could be removed and continuous (right) optimizing simply supported beam, the box is to be filled to 50% by material. Adopted from [19].

Chapter 4

Numerical Implementation

Having introduced the theoretical principles of EIM and structural optimization techniques, their implementation in a MATLAB code is reviewed within this chapter. Initially the discretization of the continuum will be briefly described and then the framework of the code will be introduced roughly in order to realize the background procedure a material, cross section and element level. A significant amount of the principle ideas of the code were adopted from Olney's code EIM-Framework which was introduced in [26], however the codes differ fundamentally in many instances.

4.1 Material description

The stress-strain relations serve for description of a constitutive law. The material behavior under tension and compression is generally described by different relations and its integral description is the foundation of the EIM. Within the code, each material is described as a structure array *Material* with function handles for implemented constitutive laws, as a function of the strain value $f(\epsilon)$. The input are the stress and strain limits, and in the case where it is not defined in the function base, the material law as a $f(\epsilon)$. The outcome from the computation are the stresses σ, specific strain energy W, F and Φ values.

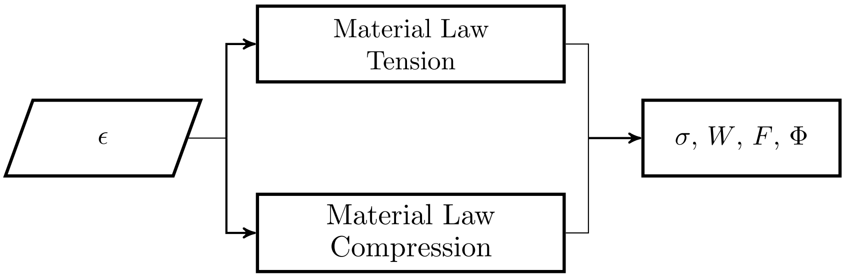


Figure 4.1: Constitutive law structure.

In Section 2.2 a linear material model was presented; however in order to take into account physical nonlinearity, the material behavior must be described by advanced functions. Constitutive models reviewed here are mainly implemented for steel and concrete, as they have been used within this work, yet the efficiency and accuracy the model, will be not discussed. Schröter et al., in [36] and [35] has in detail researched the application of steel and concrete material laws within the EIM, also taking into account long-term behavior of the concrete.
A multi-linear material law and its integral description is shown on Figure 4.2. With this material model, the behavior for significant amount of materials could be described. An example is the bilinear representation of reinforcing steel in the Eurocode [1] or the tension part of the concrete including tension softening.

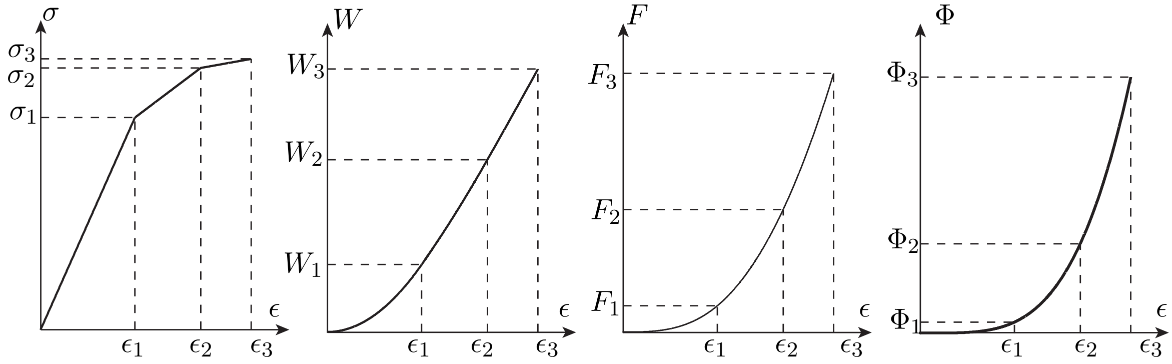


Figure 4.2: Integral description of multi-linear material law.

The behavior of concrete in compression is highly nonlinear and therefore it presents a difficulty to describe it with a multi-linear material law. In Eurocode for design of cross section, a parabolic rectangular relation is defined:

$$\sigma_c = \begin{cases} f_{cd}\left[1-\left(1-\frac{\epsilon_c}{\epsilon_{c2}}\right)^n\right], & 0 \le \epsilon_c \le \epsilon_{c2}, \\ f_{cd}, & \epsilon_{c2} \le \epsilon_c \le \epsilon_{cu2}, \end{cases} \tag{4.1}$$

where, the c stands for compression, f_{cd} is the design compression strength, ϵ_{c2} is the strain at reaching maximum strength and ϵ_{cu2} the ultimate strength. The n exponent is an empirical value and along with the other parameters could be obtained in the Eurocode. The analytical integration of the function is done by Raue in [30]. In this case it is done using analytical integration in MATLAB and as proven later, the results did not differ. On Figure 4.3 the integral description is presented. With the integration, the W and F functions are always positive regardless whether is compression or tension, which is adequate due to the fact that physically, the specific energy can not be negative.

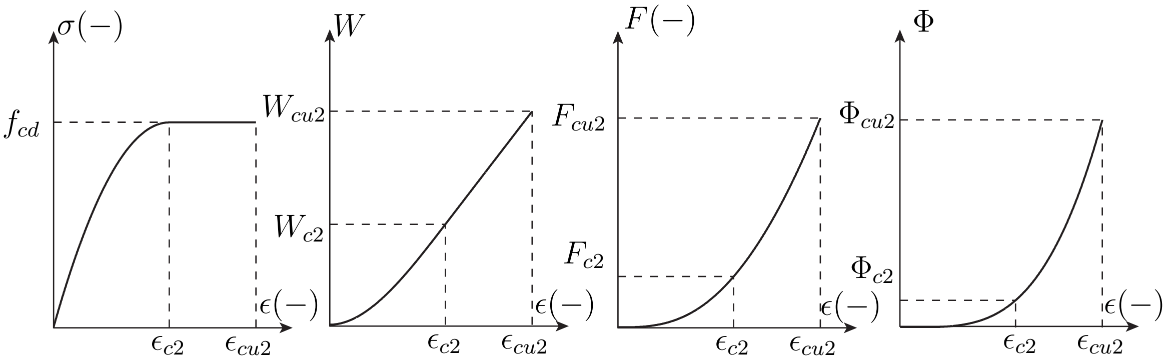


Figure 4.3: Integral description of concrete parabolic rectangular material law.

For nonlinear structural analysis and short term uniaxial loading Eurocode provides relation:

$$\frac{\sigma_c}{f_{cm}} = \frac{k\eta - \eta^2}{1 + (k-2)\eta}, \tag{4.2}$$

where, f_{cm} denotes the mean compressive strength, $\eta = \epsilon_c/\epsilon_{c1}$ for ϵ_{c1} being the peak strain and $k = 1.05E_{cm}|\epsilon_{c1}/f_{cm}|$ for E_{cm} the being mean Young's modulus. Olney in [26] states that this material model used within the EIM is bound to have issues for relatively small strains and is singular for a certain concrete strength; therefore it should be interpolated by a polynomial. The same procedure was use here also i.e. the material model was interpolated by 5^{th} order Lagrange polynomial. Figure 4.4 presents the integral description of the interpolated material model.

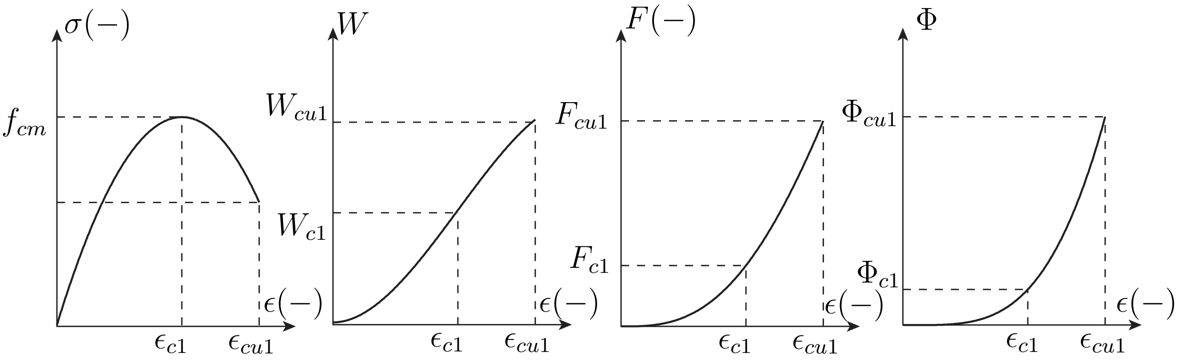


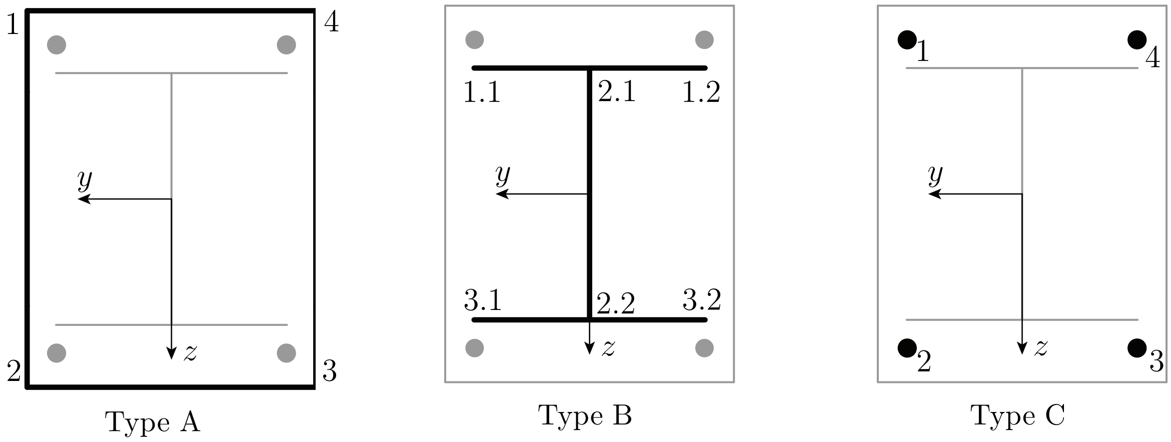
Figure 4.4: Integral description of Eurocode nonlinear concrete material law.

This material model is almost identical to the one proposed in the Model Code [2]. There exists numerous concrete models, form which commonly used are the ones by Kent and Park [27], for confined concrete by Thorenfeldt etc, which are not implemented within the scope of this work; however the material function database of the code is effortlessly modified and they could be implemented.

For the numerical implementation, the author proposes, after any material law failure criterion a relatively soft slope of the stresses to be used. This is numerically necessary for loads close or above the load limit. If a brittle material law is used, the specific energy results to a constant value and the total potential energy is not a convex function anymore. This will be thoroughly discussed in Section 4.4.1.

4.2 Cross section formulation

Previously in Section 2.2.1 the potential energy on a cross section level was introduced through line integral. Numerically, the integral is written in a form of a sum. Dependent on the strain distribution through the area, 3 types of cross section are defined: A, B and C. Polygonal type of cross section, where the strains are distributed through the both directions of the area are defined as type A, e.g. concrete sections. Cross sections which strains are distributed linearly over the length and are constant along the thickness, as a result of high length/thickness ratio, are reffed as type B, e.g. steel plates. The last one, type C, defines cross sections with constant strain distribution along the area i.e. point cross sections, e.g. reinforcement bars. These are displayed on Figure 4.5. The derivation of the numerical implementation for cross sections is adopted from Raue [30].



Type A

Type B

Type C

Figure 4.5: Types of cross section defined for the EIM.

4.2.1 Cross section type A

Internal potential energy

Having n vertices $\mathbf{P}(\mathbf{y},\mathbf{z}) = [P_1(y_1,z_1), P_2(y_1,z_1), \ldots, P_n(y_1,z_1)]$, which are defined counterclockwise for the selected coordinate system within this implementation, a piecewise linear contour contour can be described:

$$\frac{\mathrm{d}\eta}{\mathrm{d}\epsilon} = \frac{\eta_{i+1} - \eta_i}{\epsilon_{i+1} - \epsilon_i} = \frac{\Delta\eta_i}{\Delta\epsilon_i}. \tag{4.3}$$

Considering (4.3), the line integral defined in equation (2.30) which integrates the energy over the area, for cases where there exists curvature and strain difference between the i and $i+1$ point, can be transformed into a sum:

$$\begin{aligned}\Pi_i^C &= -\frac{1}{\kappa}\oint\limits_S F\,\mathrm{d}\eta = -\frac{1}{\kappa}\sum_{i=1}^{n}\int_{P_i}^{P_{i+1}} F\,\mathrm{d}\eta \\ &= -\frac{1}{\kappa}\sum_{i=1}^{n}\int_{\epsilon_i}^{\epsilon_{i+1}} F(\epsilon)\frac{\Delta\eta_i}{\Delta\epsilon_i}\,\mathrm{d}\epsilon \\ &= -\frac{1}{\kappa}\sum_{i=1}^{n}\Delta\eta_i\frac{\Delta\Phi_i}{\Delta\epsilon_i} = \sum_{i=1}^{n}\Pi_{i,i}, \qquad (\kappa\neq 0, \Delta\epsilon_i\neq 0).\end{aligned} \tag{4.4}$$

In case the strain level between the points is constant and curvature exist, the limit of the infinite small increment of the Φ function can be defined with its numerical derivative which yields to the F function:

$$\lim_{\Delta\epsilon_i\to 0} = \frac{\Delta\Phi_i}{\Delta\epsilon_i} = \frac{F_{i+1}+F_i}{2} = F_m. \tag{4.5}$$

Where the m stands for the average of the value at the i^{th} and the $i+1^{th}$ point. Thus the equation yields to:

$$\Pi_i^C = -\frac{1}{\kappa}\Delta\eta_i F_m = -\frac{1}{\kappa}\Delta(\eta F)_i, \qquad (\kappa\neq 0, \Delta\epsilon_i\neq 0). \tag{4.6}$$

If there exists no curvature, then the strain distribution is constant throughout the section as well as the specific strain energy $W(\epsilon)$, which transform the double integral in (2.23) into a line integral over the area only:

$$\Pi_i^C = W_m\iint\limits_A \mathrm{d}y\,\mathrm{d}z = \frac{W_m}{2}\oint\limits_S (y\,\mathrm{d}z - z\,\mathrm{d}y) = \frac{W_m}{2}\sum_{i=1}^{n+1}(y_i z_{i+1} - y_{i+1}z_i), \qquad (\kappa = 0). \tag{4.7}$$

It should be noted that the sum is defined up to a $n+1$ point. The supplementary point P_{n+1} is identical to P_1. Gathered, the previous relations define the internal potential energy for cross section type A:

$$\Pi_{i,i}^{C,A} = \begin{cases} -\frac{1}{\kappa}\Delta\eta_i\frac{\Delta\Phi_i}{\Delta\epsilon_i}, & (\kappa\neq 0, \Delta\epsilon_i\neq 0), \\ -\frac{1}{\kappa}\Delta(\eta F)_i, & (\kappa\neq 0, \Delta\epsilon_i = 0), \\ \frac{W_m}{2}(y_i z_{i+1} - y_{i+1}z_i), & (\kappa = 0). \end{cases} \tag{4.8}$$

Internal Forces

For sake of brevity, for the numerical implementation of the internal forces, only the final relation are presented, which are obtained from [29], wherein they are derived. The corresponding internal forces for discrete case are:

$$N = \sum_{i=1}^{n} N_i \quad M_y = \sum_{i=1}^{n} M_{y,i}, \quad M_z = \sum_{i=1}^{n} M_{z,i}. \tag{4.9}$$

From equations (2.34) for the discretized case the internal forces for type A cross section:

$$N_i^A = \begin{cases} -\dfrac{1}{\kappa}\Delta\eta_i \dfrac{\Delta F_i}{\Delta\epsilon_i}, & (\kappa \neq 0, \Delta\epsilon_i \neq 0), \\ -\dfrac{1}{\kappa}\Delta\eta_i W_m, & (\kappa \neq 0, \Delta\epsilon_i = 0), \\ \dfrac{\sigma_m}{6}(y_i z_{i+1} - y_{i+1} z_i), & (\kappa = 0). \end{cases} \tag{4.10a}$$

$$M_{y,i}^A = \begin{cases} -\dfrac{1}{\kappa}\Delta\eta_i \dfrac{\Delta(Fz)_i}{\Delta\epsilon_i} + \dfrac{1}{\kappa^2} C_i \dfrac{\Delta\Phi_i}{\Delta\epsilon_i}, & (\kappa \neq 0, \Delta\epsilon_i \neq 0), \\ -\dfrac{1}{\kappa}\Delta\eta_i (Wz)_m + \dfrac{\kappa_z}{\kappa^3}\Delta\eta_i F_m, & (\kappa \neq 0, \Delta\epsilon_i = 0), \\ \dfrac{\sigma_m}{2}(y_i z_{i+1} - y_{i+1} z_i)(z_{i+1} + z_i), & (\kappa = 0). \end{cases} \tag{4.10b}$$

$$M_{z,i}^A = \begin{cases} -\dfrac{1}{\kappa}\Delta\eta_i \dfrac{\Delta(Fy)_i}{\Delta\epsilon_i} + \dfrac{1}{\kappa^2} D_i \dfrac{\Delta\Phi_i}{\Delta\epsilon_i}, & (\kappa \neq 0, \Delta\epsilon_i \neq 0), \\ -\dfrac{1}{\kappa}\Delta\eta_i (Wy)_m + \dfrac{\kappa_y}{\kappa^3}\Delta\eta_i F_m, & (\kappa \neq 0, \Delta\epsilon_i = 0), \\ \dfrac{\sigma_m}{2}(y_i z_{i+1} - y_{i+1} z_i)(y_{i+1} + y_i), & (\kappa = 0). \end{cases} \tag{4.10c}$$

where the coefficients C_i and D_i are defined as:

$$C_i = \frac{\kappa_z}{\kappa}\Delta\eta_i - \frac{\kappa_y}{\kappa}\Delta\zeta_i + \frac{\Delta\eta_i}{\Delta\zeta_i}\Delta z_i, \tag{4.11a}$$

$$D_i = \frac{\kappa_y}{\kappa}\Delta\eta_i + \frac{\kappa_z}{\kappa}\Delta\zeta_i + \frac{\Delta\eta_i}{\Delta\zeta_i}\Delta y_i. \tag{4.11b}$$

4.2.2 Cross section type B

Internal potential energy

Consider a partition i of a type B cross section defined by points $P_{i,1}(y,z)$ and $P_{i,2}(y,z)$ with thickness b_i which is relatively small related to the length l_i. Thus the area for the integration is related to a single parameter and the line integral (2.30) yields to the numerical solution:

$$\Pi_{i,i}^{C,B} = \begin{cases} \dfrac{\Delta F_i}{\Delta \epsilon_i} A_i, & (\Delta \epsilon_i \neq 0), \\ W_m A_i, & (\Delta \epsilon_i = 0). \end{cases} \tag{4.12}$$

The area of the partition is $A_i = b_i l_i$.

Internal forces

The corresponding internal forces from the numerical partial differentiation of equation (2.34) are:

$$N_i^B = \begin{cases} \dfrac{\Delta W_i}{\Delta \epsilon_i} A_i, & (\Delta \epsilon_i \neq 0), \\ \sigma_m A_i, & (\Delta \epsilon_i = 0). \end{cases} \tag{4.13a}$$

$$M_{y,i}^B = \begin{cases} \left(\dfrac{\Delta (Wz)_i}{\Delta \epsilon_i} - \dfrac{\Delta F_i}{\Delta \epsilon_i} \dfrac{\Delta z_i}{\Delta \epsilon_i} \right) A_i, & (\Delta \epsilon_i \neq 0), \\ \sigma_m z_m A_i, & (\Delta \epsilon_i = 0). \end{cases} \tag{4.13b}$$

$$M_{z,i}^B = \begin{cases} \left(\dfrac{\Delta (Wy)_i}{\Delta \epsilon_i} - \dfrac{\Delta F_i}{\Delta \epsilon_i} \dfrac{\Delta y_i}{\Delta \epsilon_i} \right) A_i, & (\Delta \epsilon_i \neq 0), \\ \sigma_m y_m A_i, & (\Delta \epsilon_i = 0). \end{cases} \tag{4.13c}$$

4.2.3 Cross section type C

Internal potential energy

For point-like dimensions of the partition i the strains and strain energy are constant over the region A_i. From equation (2.23) it is obtained:

$$\Pi_{i,i}^{C,C} = \iint\limits_A W(y,z)\,\mathrm{d}y\,\mathrm{d}z = W_i \iint\limits_A \mathrm{d}y\,\mathrm{d}z = W_i A_i. \tag{4.14}$$

Internal forces

With the same procedure defined in the previous section, the internal force are computed, which are the same compared to ones from the basic theory of mechanics:

$$N_i^C = \sigma_i A_i, \tag{4.15a}$$

$$M_{y,i}^C = \sigma_i z_i A_i, \tag{4.15b}$$

$$M_{z,y}^C = \sigma_i y_i A_i. \tag{4.15c}$$

4.2.4 Code implementation

In order to utilize the advantages of MATLAB the code was structured to use multiple functions for calculation using structural arrays for data storage inside objects. Each step in the calculation of the total potential energy represent a function which is called during the procedure. This allows decreasing of the variable work-space which saves memory and due to the computationally cheap function call, makes the computation efficient. Memory allocation is used within initiation of the code outside the procedure for calculation, within the pre-optimization step, since it is called multiple times during optimization. The main structures for the cross section calculation are the *Material* and *SubCrossSection* structures. Containing previously defined or analytically calculated unknown function handles for constitutive laws for different materials the material structure is a nested multiple structured array which is only called during constitutive law implementation. The cross section is represented as a structural array of subcross sections which is also a structural array containing the geometrical input parameters and material assignment, as shown on Figure 4.6, where after the computation, most of the calculated parameters are stored. Generally, the code for cross section calculation is created with a modular structure which allows relatively easy modification and improvement. Within the code the cross sections of a different type forming one cross section are denoted as *subcross sections*. The procedure of calculation of the total potential energy on level of a cross section is calculated is shown on Figure 4.7. The computation of internal forces after the energy minimization, is done with the solution vector of the deformations with a similar procedure as the calculation of the internal energy with the difference that instead of calculating the energy, functions for calculation of internal forces are used.

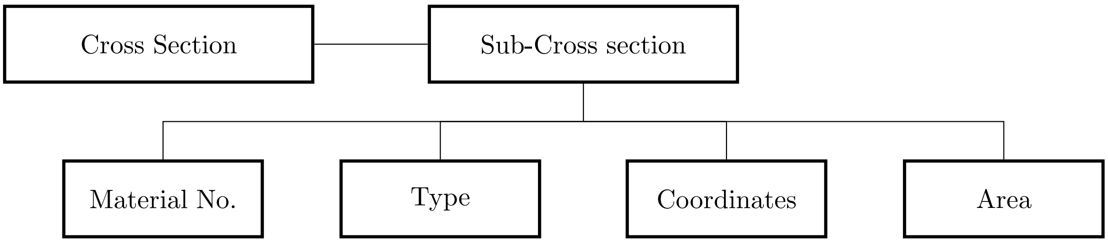


Figure 4.6: Cross section input structure.

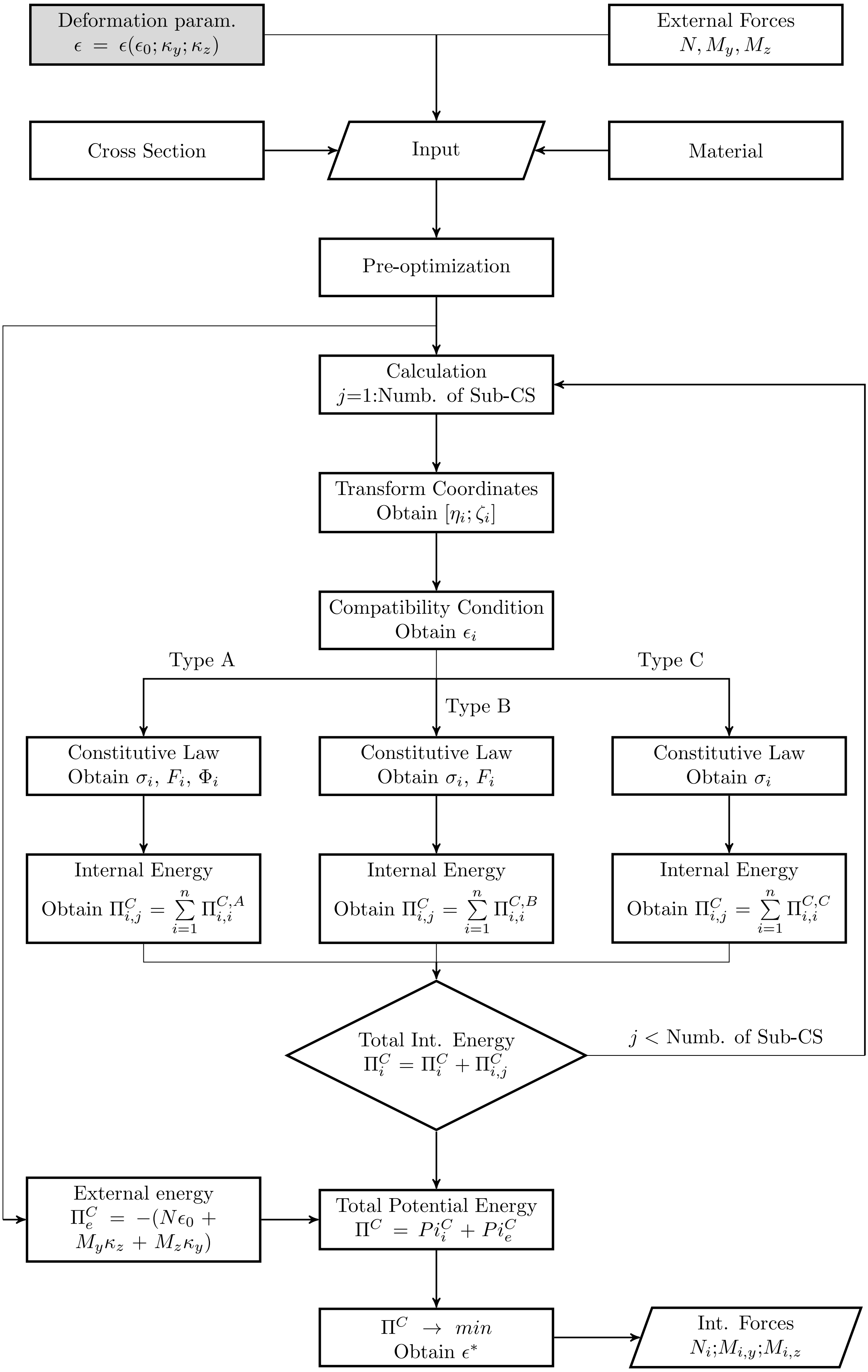


Figure 4.7: Calculation model for minimization of the total potential energy on cross section level (gray fields are the unknown variables).

4.3 Element formulation

4.3.1 Discretization

The condition for numerical implementation of EIM is to discretize the system in partitions or *finite elements* E_i for $i = 1, \dots, m$, by which the displacement functions $u(x)$, $v(x)$ and $w(x)$ and their derivatives are described in discrete points, which represent the unknown parameters within this formulation.
Generally there are two possible formulations:

- finite difference method
- shape functions

The finite difference method approximate the deformations at the nodes using the central difference quotients or Taylor's theorem enhanced central difference quotients from the displacements and rotations at discrete nodes N_i for $i = 1, \dots, p$, while the shape functions approximate the displacements and its derivatives through the whole domain of the partition, from which the deformations could be effortlessly obtained at any point. Raue et al. in [16] Schröter in [36] and Olney in [26] discuss the both methods and its advantages and disadvantages, from which it could be deducted that the finite difference method, beside its easy implementation, is generally disadvantageous due to the weak boundary condition definition and the approximation of the continuous external load in nodes. Thus, within this work only the shape functions for the discretization method are used, and the proposals of the shape functions and its implementations are from the cited sources.
As an adequate approximation for the displacements the following polynomial interpolation functions are used:

- Lagrange polynomials with C^0 continuity with $deg(f(x)) = n - 1$ using only the function values at the defined points n, $f(x_i)$ for $i = 1, \dots, n$.
- Hermite polynomials with C^1 continuity with $deg(f(x)) = 2n - 1$ using the function values and its derivatives at the defined points n, $f(x_i)$ and $f'(x_i)$ for $i = 1, \dots, n$.

On Figure 4.8 an example of discretization of a beam and interpolation of the transverse displacements with 5^{th} order Hermite polynomial is displayed:

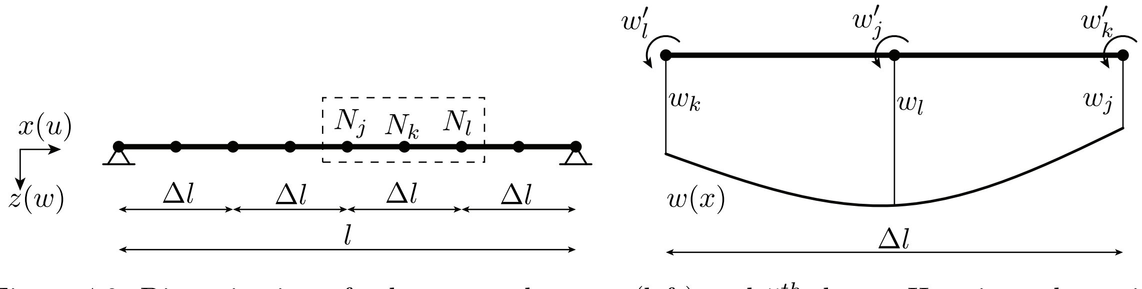


Figure 4.8: Discretization of a beam on elements (left) and 5^{th} degree Hermite polynomial approximating transverse displacements (right).

The Lagrange polynomials are used for longitudinal displacements $u(x)$ due the C^0 continuity. However, this could not be implemented for the transverse displacement $v(x)$, $w(x)$ since its derivative i.e. the rotation $v'(x)$ and $w'(x)$ is continuous throughout the element; therefore, the Hermite polynomials are used. The number of unknowns per element depends of the degree of polynomial used. Since the second derivative of the transverse displacement $v''(x)$ and $w''(x)$ describes the curvature κ_y and κ_z and the first derivative of the longitudinal one, u', the strain ϵ on cross section level, Schröter in [36] relates these two properties and proposes that the choice of polynomial for longitudinal displacements is one degree lower of the one for the transverse displacements. Within this work, two types of elements for discretization are used:

- 2Node element, using 2^{nd} order Lagrange polynomials for $u(x)$ and 3^{rd} order Hermite polynomials for $v(x)$ and $w(x)$.
- 3Node element, using 4^{th} order Lagrange polynomials for $u(x)$ and 5^{th} order Hermite polynomials for $v(x)$ and $w(x)$.

The interpolation functions and the finite elements are included in the Appendix A.
In the case of frame structures, with the assumption of rigid connection of the elements in the frame corners, the displacement and rotation are transformed with the respective transformation matrices. Here, only 2D frames are considered i.e. only the coordinate transformation matrix is used, by transforming the unknowns parameters in the connecting nodes from global $[x_g, z_g]$ to local coordinate systems $[x_i, z_i]$, where $i = 1, \ldots, m$ for m being number of elements. Due to the code transparency, the case of 3D could be implemented easily.

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} \cos\alpha_i & -\sin\alpha_i \\ \sin\alpha_i & \cos\alpha_i \end{bmatrix} \begin{bmatrix} u_g \\ v_g \end{bmatrix} \tag{4.16}$$

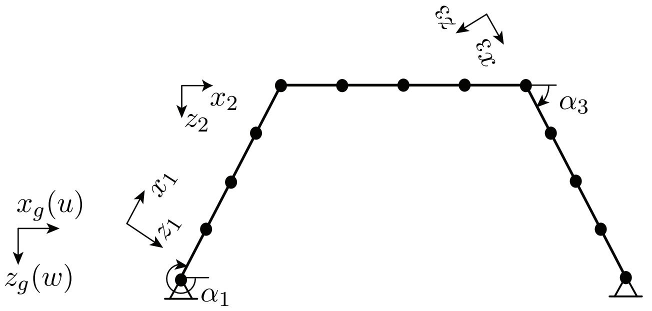


Figure 4.9: Discretization of a frame on elements with local coordinate systems.

4.3.2 Integration of the potential energy

Approximating the rotations and displacements with the interpolation polynomial, their distribution throughout the element is obtained; thus the deformation can be computed using the compliance of the compatibility equations (2.35) which permits the internal energy Π_i^E to be determined at any point of the element. Employing numerical integration rules the solution of the integral in equation (2.37) is solved which

provide the internal energy of the element. There exists numerous methods which performs numerical integration, from which in the case of the EIM two are commonly used the Gauss and Lobatto rules. The first one performs exact integration with one less integration point then the latter one, however the major disadvantage is that the location of the integration points is not in the nodes. In this case, the Lobatto quadrature is used for computing the strain potential and the external energy. The rule is exact for polynomials of degree $deg(f(x)) = 2n_p - 1$ where n_p is the number of integration points. In Appendix A the integration rule is derived for $2, 3, 4, 5$ and 6 points, as it was used within the code.
As a consequence of the previous statement, Schröter in [36] investigates the choice of the n_{ip} for the calculation of the strain energy and concludes that it depends on the degree of polynomials used for the interpolation functions, the function of the constitutive law and order of analysis i.e. whether the geometrical nonlinearity is included. Generally, using higher functions for the material law and conducting geometrical nonlinear analysis requires dense discretization, and in this case less elements with higher degree polynomial has proven to be more efficient than more elements with lower degree polynomial. According to the author, there is no direct correlation between the integration of the energy on cross section and element level, due to the versatility of material laws, i.e. one section might be yielding and other not; therefore the choice n_{ip} is intuitive. However, Schröter correlates the integration on the both levels and some of his suggestions are included in Table 4.1, which also with respect to the intuitive choice are proven to be sufficient for approximating the energy function on element level up to 9^{th} degree polynomial. The n_{ip} suggested for the previously defined elements are found in the table below:

Material law	Analysis	2Node element	3Node element
Linear Elastic	Geometrically linear	3	3
	Geometrically non-linear	4	4
Nonlinear approx. 3^{rd} order polynom.	Geometrically linear	4	5
	Geometrically non-linear	5	6

Table 4.1: Number of suggested integration points n_{ip} for Lobatto integration rule of internal energy Π_i^E (note: the nonlinear material law can be described with at least 3^{rd} order polynomial; however higher order of the approximating polynomial could also be employed).

In case of discrete forces acting on the nodes, the external energy is defined with a sum equation (4.17). When distributed load is acting on the element the external energy Π_e^E is a continuous function and also be integrated over the element, according to (2.38).

$$\Pi_e^E = -\sum_{i}^{i=n_{ep}} [P_{i,x}u_i + P_{i,y}v_i + P_{i,z}w_i + M_{i,y}\varphi_{y,i} + M_{i,z}\varphi_{z,i}]. \tag{4.17}$$

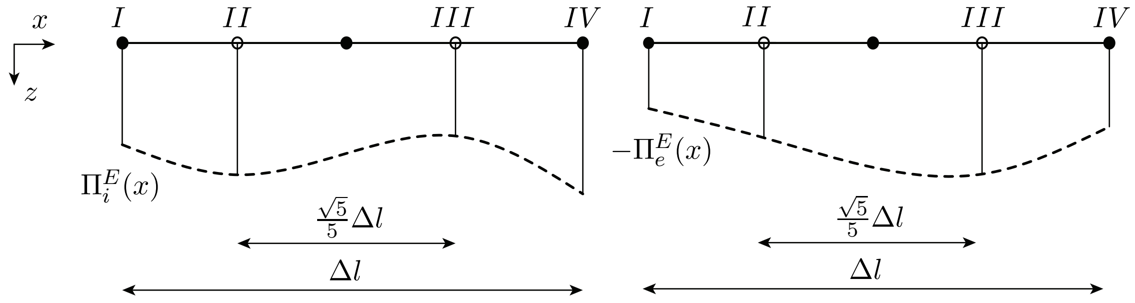


Figure 4.10: Approximations of internal $\Pi_i^E(x)$ and external $\Pi_e^E(x)$ potential energy on element level. Integration points (○); nodes (●).

The approximation of the external energy depends on the loading function, which in general case could be arbitrary and it is integrated over the element separately from the internal one. Commonly loading within civil engineering structures could be approximated ether by a constant or linear functions; thus these are implemented within the code. The number of integration points for the external energy n_{ep} could be determined exactly with respect to the interpolation function of the displacements for these loading functions. The n_{ep} for external energy are given in Table 4.2, disregarding the fact that the interpolation polynomial for longitudinal displacements is one order lower than the transverse ones for unification.

Loading function	2Node element	3Node element
Constant	3	3
Linear	3	4

Table 4.2: Number of suggested integration points n_{ep} for Lobatto integration rule of external energy Π_e^E for continuous loading.

4.3.3 Code implementation

The code for element level does not differ principally from the cross section one, rather is its extension. The main difference comes from the multiple choices for calculation, element and loading properties which could be made:

- 2D or 3D analysis with respect to the loading,
- continuously loaded elements or only node loads included,
- 3 or 2 node or truss finite element used,
- geometrically linear or non-linear analysis,
- number of integration points for internal and external energy.

To increase computational efficiency function handles are created within the preoptimization stage without any additional effort for input. This being done, the number of unknown variables for the optimization are decreased with the removal of inactive degrees of freedom, additional procedures are avoided, for instance the external energy is integrated in the direction only if there is prescribed loading on the element and therefore the shape functions for the displacement computation in the other directions and elements are idle. Memory allocation is also included within this stage. The implemented structures *Element* and *Node*, are displayed below:

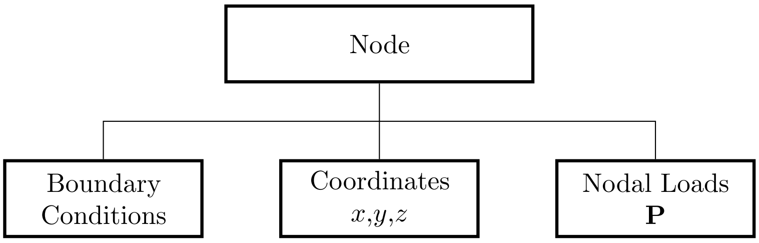


Figure 4.11: Node input structure.

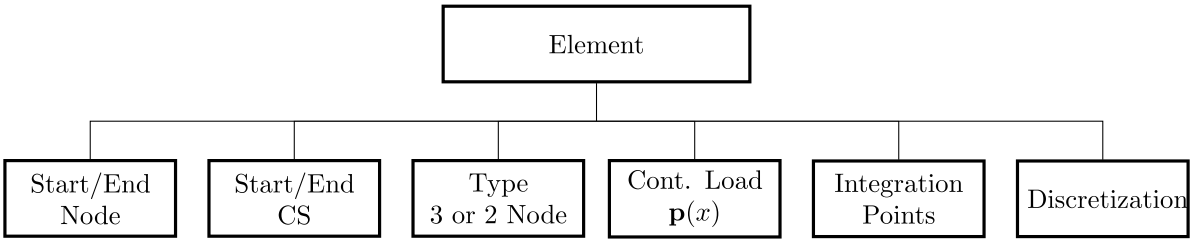


Figure 4.12: Element input structure.

The *Element* is initially used only for input and later is reformulated in *ElementS* structure array. This step was taken in order to simplify the input and create a user-friendly code. In a fact the latter structure presents the term finite element defined in the previous section. Figure 4.13 represents the flowchart of the calculation scheme for the total energy of the system. As an addition to the function handles, parameters which are constant for the optimization process, are calculated in the pre-optimization stage such as the projected load values in each integration point $p_{l,j}$ for $j = 1, \dots, n_{ep}$, transformation matrices, lengths of elements etc. The external energy is divided within two parts, in the nodes Π_e^N and in each element $\Pi_{e,k}^{E,c}$ for $k = 1, \dots, m$, which allows the first one to be excluded of the element loop and latter one calculated within. After realizing the change in the vector of unknowns parameters, within the optimization build-in solvers in MATLAB, occurs only partially in some members Olney in [26] proposed a solution within the optimization to identify them and compute the elements which are associated with them, by storing the idle ones in persistent variables. Therefore, the loop is only for the *Active Elements* which allows significant reduction of computational time. Within the Lobatto integration rule, w_j are defined weighting functions depending on number of integration points n_p; thus the parameters required for the calculation of the internal energy, the deformation parameters ϵ, and the ones required for external energy from continuous loads, the local discrete displacement vector $\mathbf{U}_l$, are computed in these points. After reaching a stationary point, the solution vector $\mathbf{U}_l^*$ is obtained, with which in the next step the computing the internal forces at the integration points with basically the same procedure displayed below.

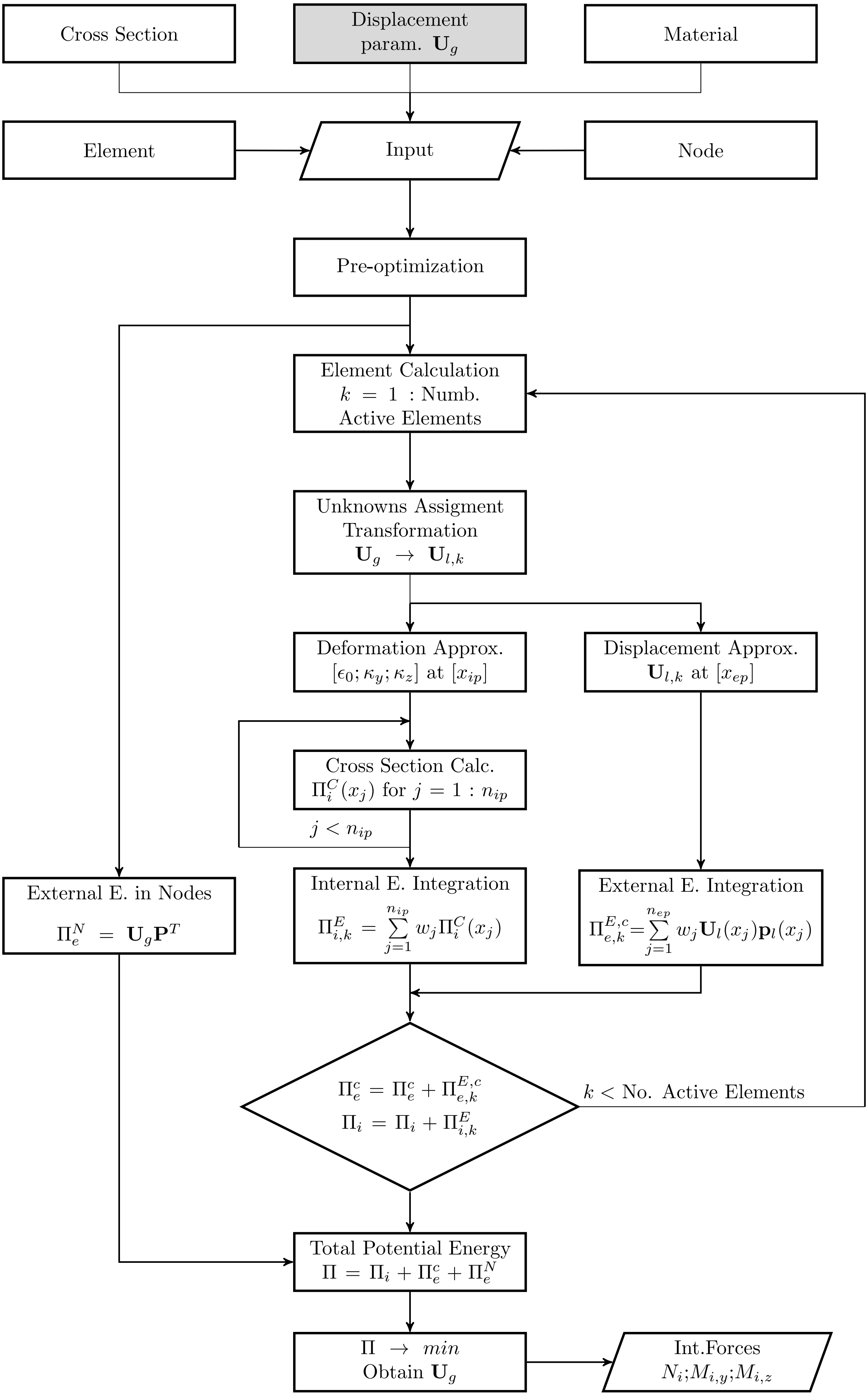

Cross Section
Displacement param. $\mathbf{U}_g$
Material
Element
Input
Node
Pre-optimization
Element Calculation $k = 1$: Numb. Active Elements
Unknowns Assigment Transformation $\mathbf{U}_g \rightarrow \mathbf{U}_{l,k}$
Deformation Approx. $[\epsilon_0; \kappa_y; \kappa_z]$ at $[x_{ip}]$
Displacement Approx. $\mathbf{U}_{l,k}$ at $[x_{ep}]$
Cross Section Calc. $\Pi_i^C(x_j)$ for $j = 1 : n_{ip}$
$j < n_{ip}$
External E. in Nodes $\Pi_e^N = \mathbf{U}_g \mathbf{P}^T$
Internal E. Integration $\Pi_{i,k}^E = \sum_{j=1}^{n_{ip}} w_j \Pi_i^C(x_j)$
External E. Integration $\Pi_{e,k}^{E,c} = \sum_{j=1}^{n_{ep}} w_j \mathbf{U}_l(x_j) \mathbf{p}_l(x_j)$
$\Pi_e^c = \Pi_e^c + \Pi_{e,k}^{E,c}$
$\Pi_i = \Pi_i + \Pi_{i,k}^E$
$k <$ No. Active Elements
Total Potential Energy $\Pi = \Pi_i + \Pi_e^c + \Pi_e^N$
$\Pi \rightarrow min$ Obtain $\mathbf{U}_g$
Int.Forces N_i;$M_{i,y}$;$M_{i,z}$

Figure 4.13: Calculation model for minimization of the total potential energy of a system.

4.4 Optimization procedure

Dealing with optimization with the EIM, the optimization procedures are used on two levels, minimizing the potential with which the internal equilibrium is established, and an objective function, by which, optimal shape or structural parameters are acquired.

4.4.1 The function of the potential energy

Before an optimization algorithm is used, the total potential is reviewed as a function. According to the elastic theory mechanics, the total potential energy is a strict convex function which could be easily solved with the gradient algorithms. The analytical solution on element level (only the unknown parameters differ on cross section level) concerning the sufficient condition after equation (3.4) could be formulated as:

$$\nabla\Pi(\mathbf{U}^*) = 0, \tag{4.18a}$$

$$(\mathbf{U}^*)^T\nabla^2\Pi(\mathbf{U}^*) > 0. \tag{4.18b}$$

Where $\mathbf{U}^*$ is the solution vector for the displacements. The total potential, besides the external forces, depends on the material law used. Using a nonlinear constitutive law will change the function space, and in case of limited or perfect plastic deformation Π becomes a non-convex function, where the solution is only a local minimum for external forces below the capacity. Applying load above the capacity, there is no solution i.e. the total potential is continuously decreasing function or a constant for perfectly plastic material. The non-convexity of the total potential is due to the constant $W(\epsilon) = W(\epsilon_y) = const$ or linear specific strain energy $W(\epsilon) = W(\epsilon_y) + \epsilon$. Figure 3.7 illustrates a bilinear steel material law in tension, where two cases are displayed, one with brittle failure and one with perfect plasticity.

On cross section level the Π_i from equation (2.23) for $W = const$ yields to:

$$\Pi_i^C = \iint\limits_A W(y,z)\,\mathrm{d}y\,\mathrm{d}z = \iint\limits_A W(\epsilon_u)\,\mathrm{d}y\,\mathrm{d}z = W(\epsilon_u)A = const, \tag{4.19}$$

and for the perfectly plastic case, for the region after the yielding:

$$\Pi_i^C = \iint\limits_A W(y,z)\,\mathrm{d}y\,\mathrm{d}z = \iint\limits_A W(\epsilon)\,\mathrm{d}y\,\mathrm{d}z = W(\epsilon_y) + (\epsilon - \epsilon_y)A. \tag{4.20}$$

This is valid in case of cross sections with constant strain distribution (Type C) and for cross sections without constant strain distribution throughout the area (Type A and B) loaded with a normal force. Theoretically, in case of uniaxial or biaxial bending for the latter cross section types, this is true only when the curvature is approaching infinity. Numerically the curvature could not reach infinity, however the slope of the internal energy after yielding is infinity small; thus it is valid to conclude that the strain energy is constant or linear. In the first case, it is clear that equilibrium is unstable due to the increasing external energy , and in the latter case the slope of the internal energy is the same as the external one, therefore the total potential is a constant function.

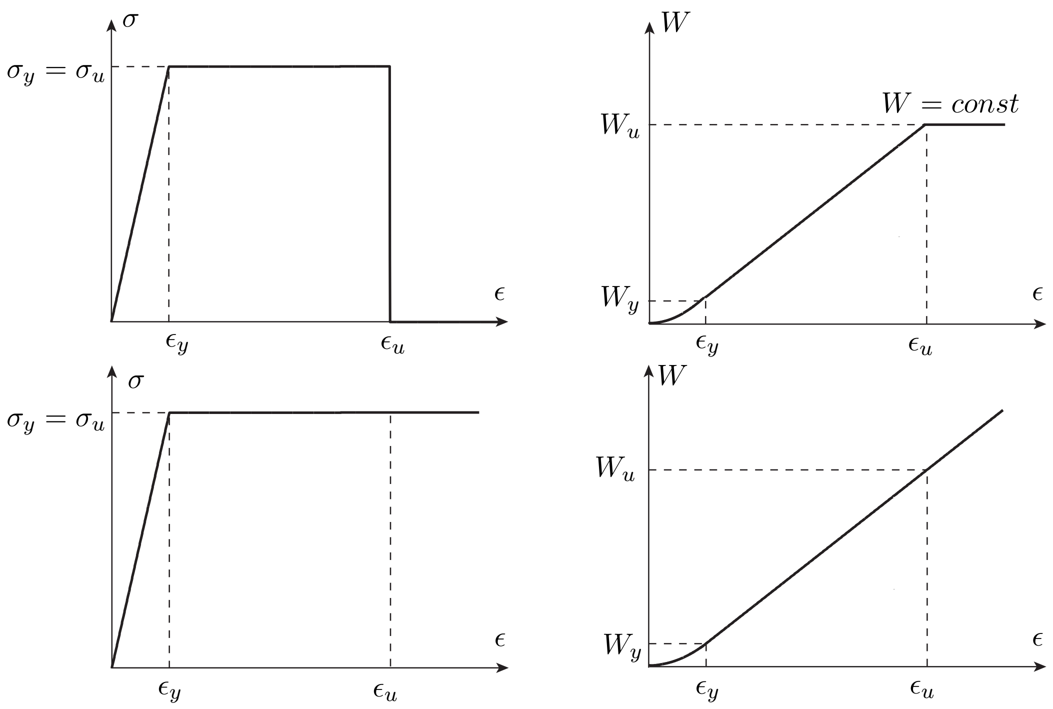


Figure 4.14: $\sigma - \epsilon$ and $\sigma - W$ relationships for brittle (top) and perfectly plastic (bottom) bilinear constitutive law.

In order to avoid the non-convexity of the total potential, which creates numerical problem, a slight numerical slope after the ultimate strain will result to a quadratic specific energy and ultimately with a stable equilibrium for any loading.

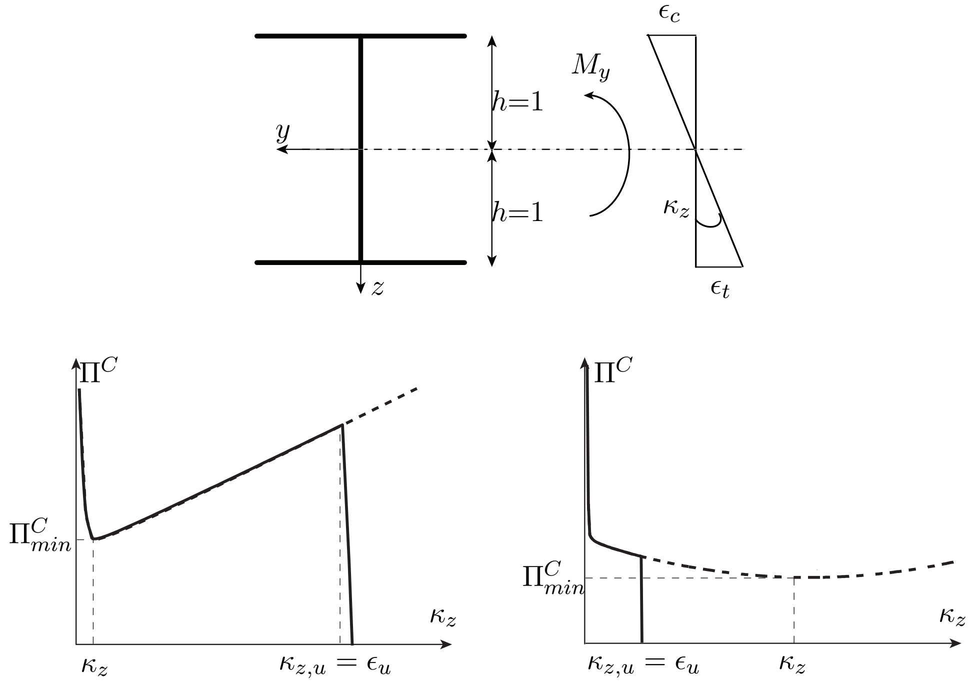


Figure 4.15: Total potential of cross section (top) with applied moment just below the capacity $M < M_{max}$ (bottom left) and just above capacity $M > M_{max}$ (bottom right) for brittle constitutive law (—) and one with numerical slope after the ultimate strength (- -).

Figure 4.15 illustrates the unstable equilibrium on cross section level with the application of a external moment just below and just above the capacity moment $M_f max$ for brittle and with a numerical slope for bilinear constitutive law. For $M \leq M_{max}$ the minimum of the potential is clear for both cases, and in the case when the loading is above the critical one, the section with brittle constitutive law fails to find a minimum i.e. it is a unstable equilibrium, while the other one, obtains a minimum for significantly greater curvature then the yield one. For geometrically linear problems, on a element level, for brittle or perfectly plastic material the unstable equilibrium occurs after the moment redistribution i.e. when there is kinematic mechanism. Introducing the geometrical non-linearity on element level, there exists another unstable equilibrium of the energy i.e in the case of buckling.

After realizing the nature of convexity of the potential, it could be established that for its minimization the best choice for algorithms would be the gradient methods. Within the code this is implemented through MATLAB optimization toolbox which offers for medium scale optimization the BFGS quasi-Newton method using line-search method for step size control within the *fminunc* function for unconstrained optimization.
Furthermore in the case of inflection point in the specific energy in the material law $W(\epsilon_{inp})$, the second derivative is equal to 0 ($W''(\epsilon_{inp}) = 0$); thus in case of external loading, for which the minimum of the potential is close to this point, may only the necessary condition shown in equation (3.4b) to be satisfied for the local minimum i.e. the Hessian is positive semidefinite and therefore the potential function is not a strict convex, but only a convex function. An example of an inflection point was observed within the nonlinear concrete compression constitutive law according to the Eurocode displayed in Figure 4.4 between the strain for compression strength ϵ_{c1} and the ultimate strain ϵ_{cu1}; therefore in case the external load is above the compression strength the potential energy has a local maximum which causes numerical problems for the structural optimization part.

4.4.2 Structural optimization procedure

The objective of this work is to implement the structural optimization problem, on a cross section and element level using EIM. For the cross sections, the equilibrium condition was proposed to be implemented in two ways: equalizing the external and internal forces with the calculation model introduced in Figure 4.7 and with multi-objective optimization using the minimum of potential energy as one objective and an additional objective function for the design variables. For the element model, only the latter one was proposed due to the moment distribution. However the author proposes that structural optimization using the energy methods is a bilevel optimization problem briefly introduced in Section 3.3, equation (3.22). Before elaborating the methods, it should be noted that in all of the cases, a material with numerical slope after the ultimate strain in order that the total potential function space is defined outside the failure criterion, as discussed in the previous section.
Using the equilibrium constrains on cross section level is a computationally cheap way of structural optimization. The formulation could be ether using a constrained single objective algorithms or unconstrained one by employing penalty functions. The formulation, with respect to the SO problem formulated in equation (3.20), within the

code procedure is given on the flowchart on Figure 4.16. The objective function and the constrains are functions created externally from the code and then assigned within the execution script.

MATLAB offers the *fmincon* function for constrained optimization using the gradient mehods. There are four algorithms which are implemented within this function, from which two are suggested to be used within problems of the EIM: *active-set* algorithm which is a Lagrangian based method *interior point* which approximates the optimization problem with linearization of barrier constrains also by Lagrangian multipliers, thus it can recover from solutions outside the function space. Generally, the active set is relatively faster in the initiation of the optimization since it takes larger steps, which is beneficial in the case of the EIM method when it is dealt with large value state variables such as maximum moment; however this is a medium-scale algorithm which stores and operates with full matrices during the optimization and if a convergence is not secured after relatively many steps, the efficiency drops significantly. On the other hand, the interior point is good when dealing with both dense and sparse problems and in case of state variables for which the potential energy is not defined, e.g. area, yet this could be also perceived as a disadvantage since it may lead to inaccuracy for stationary values close to the constrains. The interior point is a large-scale algorithm meaning it does not store nor use full matrices during the optimization which is beneficial using many variables. These methods in this case are used intuitively.

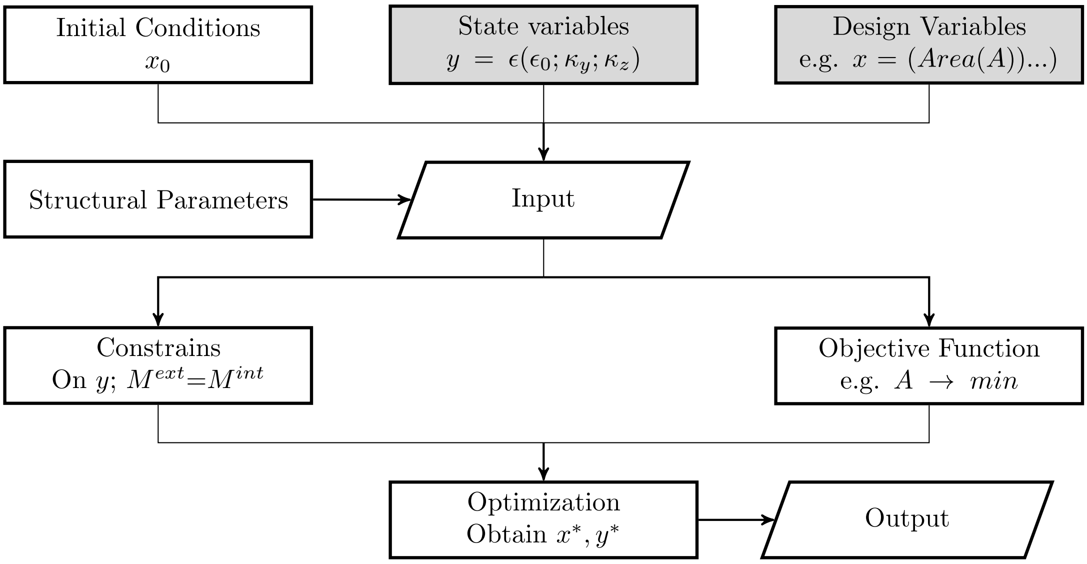


Figure 4.16: Calculation model for optimization on cross section using equilibrium conditions as constrains (gray fields are unknown parameters).

Since the previously described SO problem could not be employed on element level, the multi-objective method with two objective functions, the total potential and other minimizing a design variable was proposed in the following formulation:

$$SO: \begin{cases} \min\limits_{\mathbf{x},\mathbf{y}} [\Pi(\mathbf{x},\mathbf{y}), f(\mathbf{x},\mathbf{y})], \\ \text{subject to} \begin{cases} G_j(\mathbf{x}, y) \leq 0, & j = 1 \dots N_g, \\ H_k(\mathbf{x},\mathbf{y}) = 0, & k = 1 \dots M_h, \\ x_i^l \leq x_i \leq x_i^u, & i = 1 \dots n, \\ y_j^l \leq y_j \leq y_j^u, & j = 1 \dots m. \end{cases} \end{cases} \tag{4.21}$$

where y and x are the state and design variables respectively. The second objective function is equal to the area $f(A) = A$. After reviewing the method through principle examples and tests, in authors opinion, its implementation is not possible. A principle structural optimization design example is presented: single reinforcement bar with unknown area to be designed for elastic strains for a certain force, presented with the constitutive law on Figure 4.17. The cross section and the forcing were chosen in order to have constant strain distribution for simplicity. The example is first solved analytically with the following values:

d	A	F	$\epsilon_0 = \epsilon_y$	σ_y	W_y	Π_i	Π_e	Π
[cm]	[cm^2]	[kN]	[‰]	[kN/cm^2]	[kN/cm^2]	[kN]	[kN]	[kN]
2.00	3.14	70.02	2.00	22.30	0.0223	0.07	−0.14	−0.07

Table 4.3: Analytical solution values for reinforcement bar subjected to axial force.

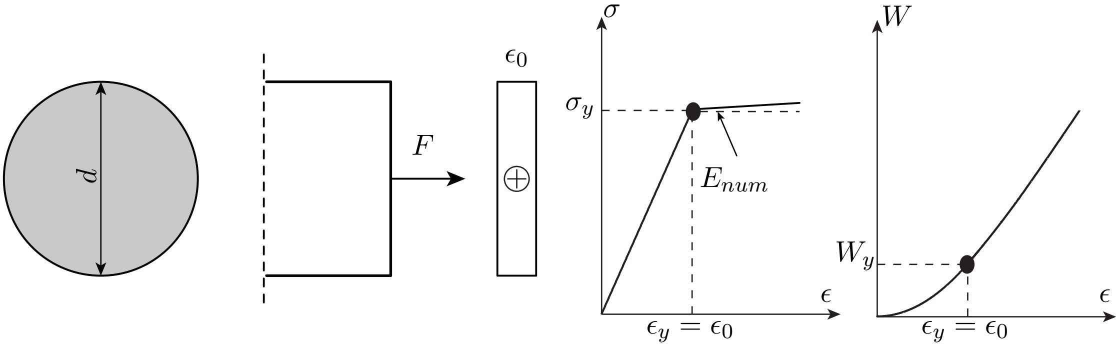


Figure 4.17: Reinforcement bar with bilinear constitutive law with small numerical slope, subjected to an axial force.

The multi-objective optimization problem, to obtain the minimum area is formulated:

$$SO: \begin{cases} \min\limits_{A,\epsilon_0} [\Pi(\epsilon_0, A), f(A)], \\ \text{subject to} \begin{cases} \epsilon_y - \epsilon_0 \leq 0‰, \\ -A < 0\text{cm}^2, \end{cases} \end{cases} \tag{4.22}$$

where the state variable is the longitudinal ϵ_0 and the design variable the area A. The constrains are the area to be positive and the strain to be less than the yielding one ϵ_y. After formulating the SO, first the objective functions are presented on Figures 4.18 and 4.19 with respect to the both variables. Naturally the area is independent of the strain, and it could be observed that the potential energy is a convex function, for which the minimum in each plane section through the area is also decreasing. The plots are up to $\epsilon=\pm$ 5‰, however there exists a minimum for the lower values of the area for lower strain, as discussed in the previous section, which are not displayed in order the plot to be clear.

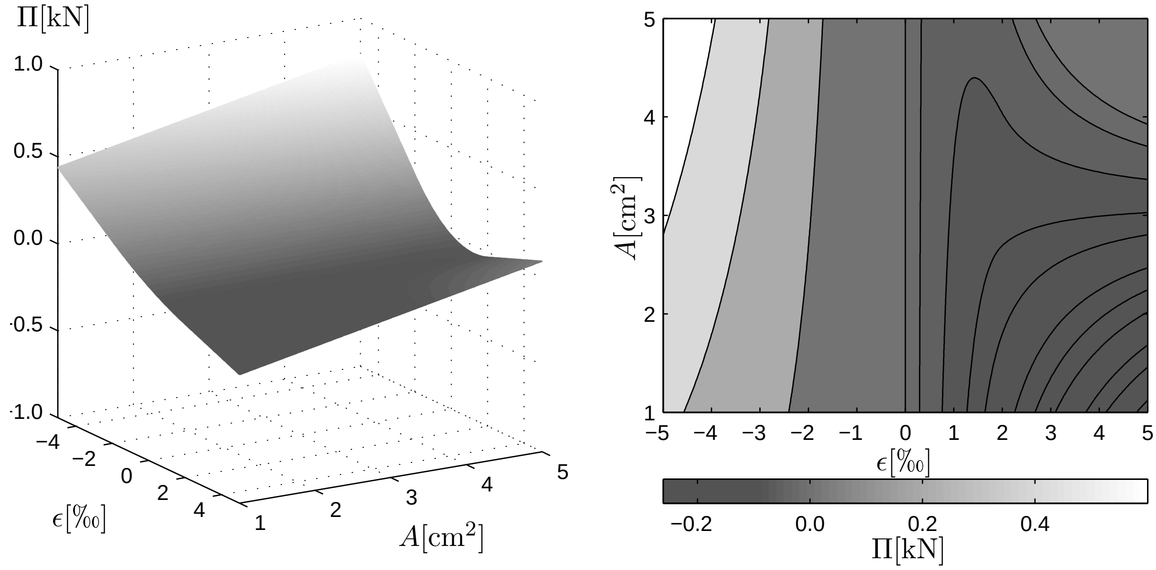


Figure 4.18: Total potential energy function $\Pi(A, \epsilon_0)$ in 2D contour and 3D plot.

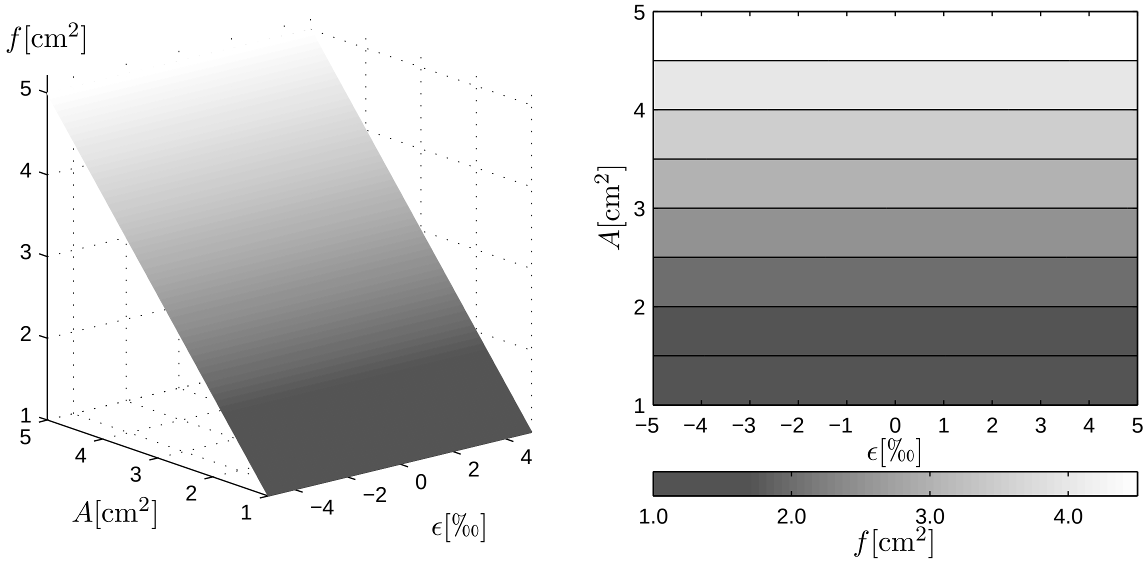


Figure 4.19: Representation of $f(A) = A$ in 2D contour and 3D plot.

It could be concluded that both objective functions are decreasing with lowering the area, normally only the potential energy is dependent on the strain. Since the material has a numerical slope after the ultimate strain there is a stationary point for each value of the area, which for lower area of the bar is for very high strains. Having the both functions unconstrained yields to continuously decreasing functions as the area decreases and strain increases i.e. there is no minimum; thus, constrains must be applied. The design constrains (the negative area) are not going to be reviewed in this case since they are easily applicable on $f(A)$ and therefore the variable A will be presented in interval [1,5] in which, from the analytical method, it is certain that the solution exists. With this, the constrains on the state variable, ϵ_0, are left to be applied, thus ether the potential or the second objective function can be constrained. For clarity the constrains will be applied in term of penalty functions. Constraining both objective functions would yield in the same result, as proved further.
Constraining the potential with a penalty function would yield in the penalized potential as objective function, for strains higher or lower then 2‰:

$$\tilde{\Pi}(A, \epsilon_0) = \Pi + \phi, \quad \text{where,} \quad \phi(\epsilon_0) = \begin{cases} 0, & |\epsilon_0| < 2‰, \\ k(|\epsilon_0| - 2)^2, & |\epsilon_0| \geq 2‰. \end{cases} \tag{4.23}$$

The parameter k is just a weighting factor, which is this case was chosen to be 0.05 with respect to the objective function value. The result is presented on Figure 4.20, where it is clear that the penalized function increases for values for ϵ_0 above the constrain, however the value is still decreasing with respect to A <3.14cm^2. The values are on the boundary of the constraint and present quasi-stationary points with respect to ϵ_0; however they are not a stationary point of the total potential, i.e. there is no equilibrium for value of the area below the solution one. This could be interpreted in a way that the lowest stationary point of the total potential within the constrains is the solution; however this can not be obtained with a multi-objective algorithm.

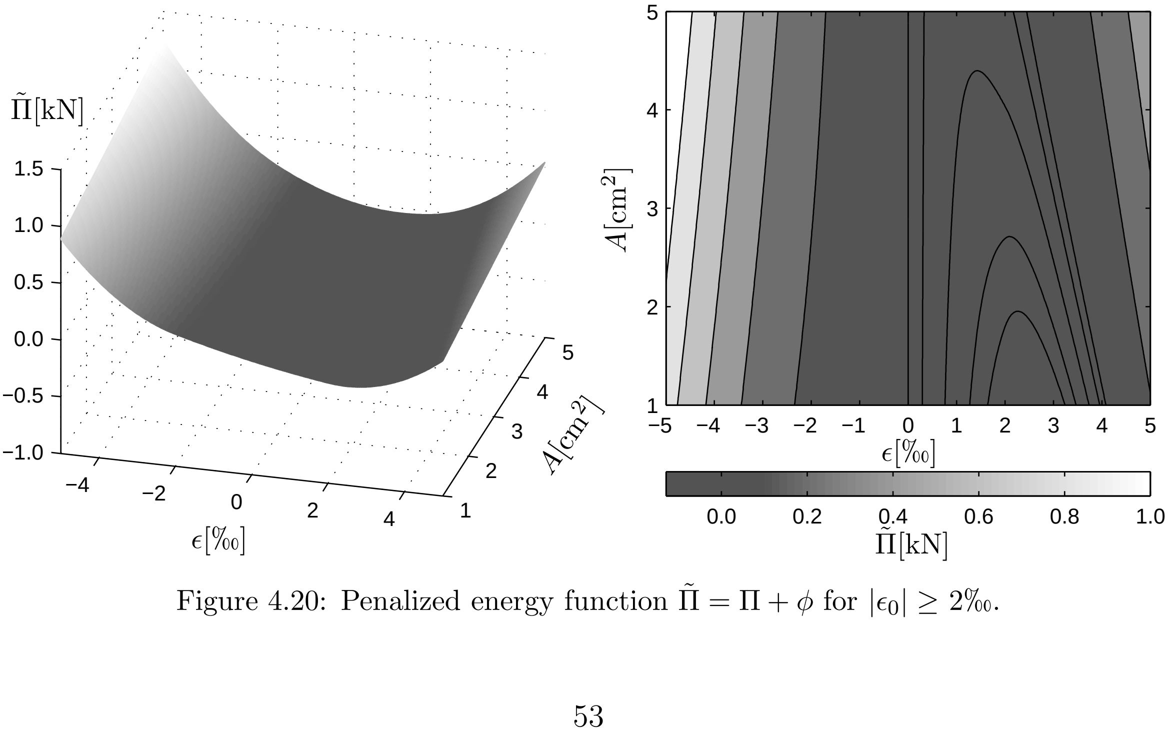


Figure 4.20: Penalized energy function $\tilde{\Pi} = \Pi + \phi$ for $|\epsilon_0| \geq 2‰$.

It should be noted, that within the previous discussion, a stationary point is meant with respect to the state variables y. This concludes that penalizing the energy is principally wrong choice and yields to results for which there is no equilibrium in the system.

The second objective function, $f(A)$, could be penalized in the same way as the total potential:

$$\tilde{f}(A, \epsilon_0) = A + \phi, \quad \text{where,} \quad \phi(\epsilon_0) = \begin{cases} 0, & |\epsilon_0| < 2‰, \\ k(|\epsilon_0| - 2)^2, & |\epsilon_0| \geq 2‰. \end{cases} \tag{4.24}$$

where, the weighting factor k in this case is chosen to be 1, according to the function values. On Figure 4.21, as it is expected the penalized function with respect to ϵ_0 is increasing after the penalized values, and it is constant in-between. However, there is no restraint of the function with respect to the value of the area below the solution, which in general case would be an unknown, and therefore it can not be constrained.

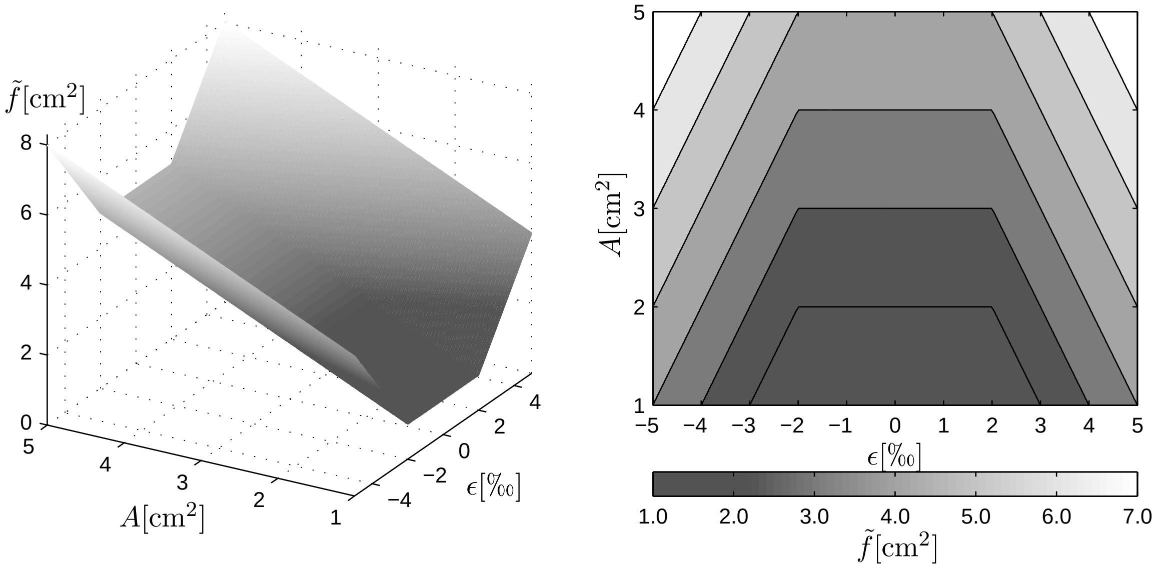


Figure 4.21: Penalized second objective function for $\tilde{f}(A, \epsilon_0) = A + \phi$ for $|\epsilon_0| \geq 2‰$.

This combined with the potential function, presented in Figure 4.18, it could be concluded that both are decreasing unrestrained after the real solution i.e. below of area A^{*}=3.14 cm^2. Therefore with the multi-objective algorithms this problems can not be solved, since in this case, only the range of the variables, in which the real solution is included, is defined i.e. the interval of the lowest stationary point of the total potential with respect to ϵ_0. This is by definition a bilinear optimization problem. Additionally, on Figure 4.22 the two functions, the potential and penalized second objective function, are shown where the design variable, the area, is defined for two cases: at the solution i.e. $A = A^{*}$ and below $A < A^{*}$. Clearly, for the latter the both function values are lower, and as a consequence any multi-objective solver will indicate as a solution. However, only for the required area the derivative of the potential function with respect to ϵ_0 is 0 within the constrains.

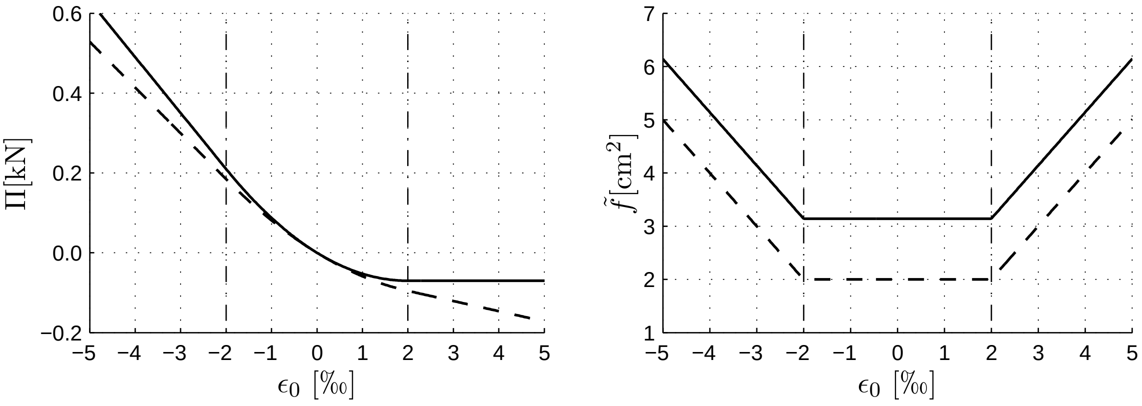


Figure 4.22: Total potential energy Π (left) and penalized second objective function $\tilde{f}$ (right) sections for the solution value of the area A =3.14 cm^2= A^* (—–) and below the required A=2.0cm^2< A^* (– –) with the indicated constrain at ±2‰(– · –).

The proposed method of structural optimization within this work is the bilinear or the nested optimization presented previously in equation (3.22). The reasons come from the essential differences between the bilevel and multi-objective optimization reviewed in Section 3.2.3, and the observations are the following:

- Multi-objective optimization problems results with multiple solution, whilst in the bilevel case there exists only one. Dealing with the energy, no compromise is allowed by decision maker's opinion, since without stationary point of the energy i.e. minimum, the equilibrium condition is not met. Unlike the multi-objective, for each $f^u(x)$ of the upper objective function, the bilevel optimization guaranties equilibrium as a result of the included stationary point of the energy within the constrains, hence Figure 3.7.

- To solve a structural optimization problem, design on x and state constrains on y must be applied , ether within the objective function as a penalty term or as external constrains within the optimizer e.g. maximum strain, stress, displacement, negative area etc. No constrains can be applied on the potential, with respect to the state variables, due to the fact that constraining a convex function may very well result into a solution on the constraint's boundary, as presented for a simple function in Figure 3.6. In terms of a penalty function this could be written as:

$$\frac{\partial(\Pi(\mathbf{y}) + f(\mathbf{y}))}{\partial \mathbf{y}} \neq \frac{\partial(\Pi(\mathbf{y}))}{\partial \mathbf{y}}. \tag{4.25}$$

- As a consequence of the previous statement, the constrains must be applied on the second objective function or the upper one in case of bilevel optimization. By applying the constrains on the second objective function, the result is a limited function space, where the solution of the structural optimization exists; however stationary can not be proven i.e. there is a lower value of the potential within this space than the lowest stationary point, due to the dependency of the potential energy with respect to the design variables. On the other hand, applying constrains on the upper objective function means constraining a function for which equilibrium conditions are met.

The lower function in this case is the potential, and the upper can be arbitrary. Constrains on the deformation or displacements (state variables) parameters could be applied after a equilibrium is reached and the design variables can be constrained independently. There exists variety of objective functions and basically each optimization problem presents a unique one, however there are ones which are usually used such as maximize the compliance, minimize the maximum stresses, displacements under defined conditions by modification of the system. The EIM method is based on the kinematic formulation and here it is mainly dealt with an upper objective function which minimize the difference between the allowable strain, with respect to the material, and the strain in the most critical point of the section, or in the system, on element level with which the maximum of the system is utilized:

$$f(\mathbf{x},\mathbf{y}) = (max(\epsilon_i - \epsilon_u^+; \epsilon_u^- - \epsilon_i))^2, \qquad i = 1, \ldots, n_p, \tag{4.26}$$

where ϵ_i is the corresponding strain of n_p number of points of the cross section, of all integration points on element level, ϵ_u denotes the ultimate strain of the corresponding constitutive law, in tension $(+)$ and in compression $(-)$. The idea of the squared term is to create a convex function, which in the best case scenario would be 0, and incorporate a penalty term for $f(\mathbf{x},\mathbf{y}) > 0$. By doing this an bilevel optimization problem is formed, unconstrained on both levels:

$$SO: \begin{cases} \min\limits_{\mathbf{x},\mathbf{y}} f(\mathbf{x},\mathbf{y}) = \min\limits_{\mathbf{x},\mathbf{y}} (max(\epsilon_i - \epsilon_u^+; \epsilon_u^- - \epsilon_i))^2, \qquad i = 1, \ldots, n_p, \\ \text{where } \mathbf{y} \text{ solves} \begin{cases} \min\limits_{\mathbf{y}} \Pi(\mathbf{x},\mathbf{y}). \end{cases} \end{cases} \tag{4.27}$$

In case there are any design constrains, it is proposed to be incorporated as a penalty function, since more optimization algorithms are implemented in MATLAB for the unconstrained one. Further more, there is higher level of control of the objective function and there exists no difficulties for weak penalty, since as discussed before, with numerical slope after ϵ_u the function space of the potential is unlimited.[1]

The implementation within the program of the upper and the lower optimization is displayed on Figure 4.23 for element level, which uses, as an example of design variables, coordinates of a point P_i. The superscript ^{o} denotes assigned parameters during the upper optimization. The most robust bilevel optimization is implemented in the code, the nested one, with which the two different optimization problems use two different algorithms. As proposed before, the lower one is a convex function and the *fminunc* function containing gradient newton algorithms is used. The upper one, although there is convexity, numerical problem appears when there is abrupt changes of the gradient. Therefore for the most examples, the simplex algorithm is commonly used within this work, which is implemented in the MATLAB in the *fminsearch* function. This function does not include constrains defined in external function therefore penalty functions were used, regardless of its weak constraint for the reasons explained above. The outer objective function is defined externally as well as the function which assigns parameters, then both are included within the calculation setup structure.

[1] only for geometrically linear systems; in case of geometrical non linearity, other restrictions apply.

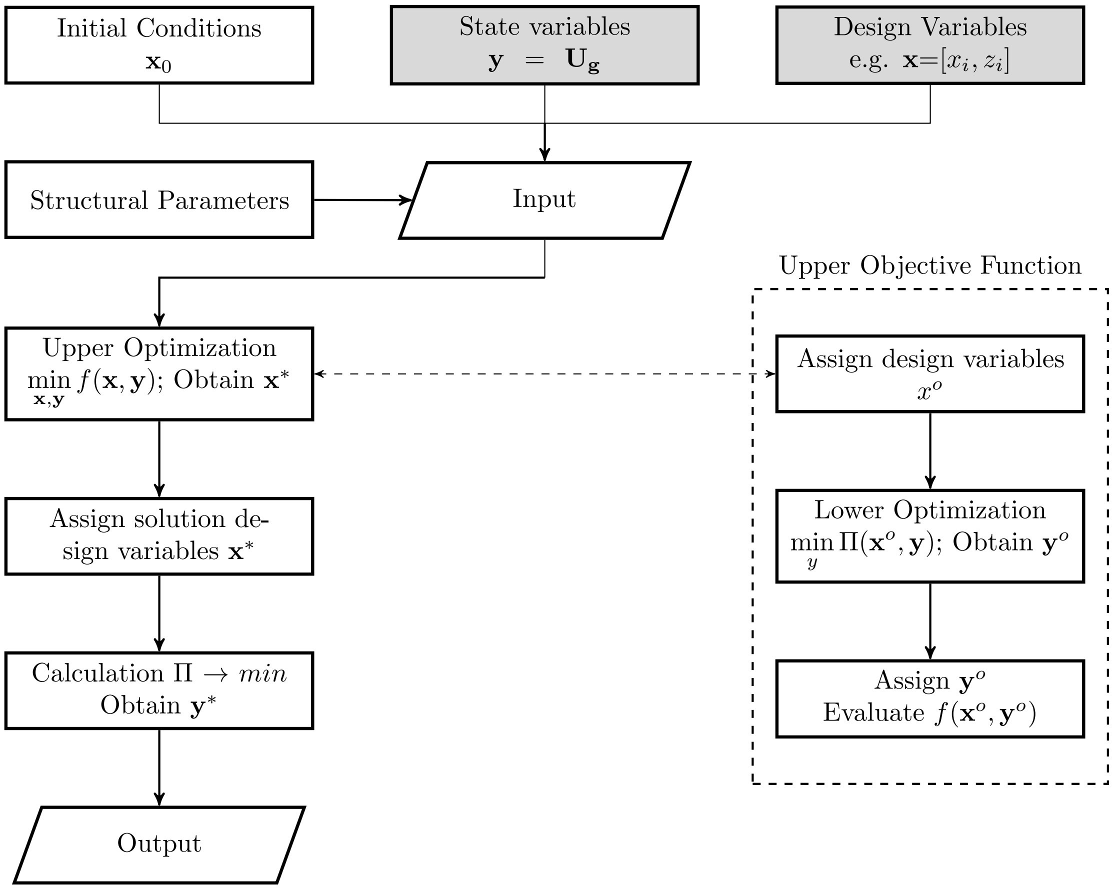

Initial Conditions $\mathbf{x}_0$
State variables $\mathbf{y} = \mathbf{U_g}$
Design Variables e.g. $\mathbf{x}=[x_i, z_i]$
Structural Parameters
Input
Upper Objective Function
Upper Optimization $\min_{\mathbf{x},\mathbf{y}} f(\mathbf{x},\mathbf{y})$; Obtain $\mathbf{x}^*$
Assign design variables x^o
Assign solution design variables $\mathbf{x}^*$
Lower Optimization $\min_{y} \Pi(\mathbf{x}^o,\mathbf{y})$; Obtain $\mathbf{y}^o$
Calculation $\Pi \rightarrow min$ Obtain $\mathbf{y}^*$
Assign $\mathbf{y}^o$ Evaluate $f(\mathbf{x}^o,\mathbf{y}^o)$
Output

Figure 4.23: Calculation model for optimization on a element using nested optimization (gray fields are unknown parameters).

In some cases, the genetic algorithm within the MATLAB function *ga* was implemented, as a consequence of the behavior of the simplex algorithm on local minimums i.e. it might not converge. However, after the optimization using the genetic algorithm, hybrid method can be employed, by introducing the solution as a initial vector for the simplex algorithm. The initial conditions are essential for every good structural optimization as discussed before. These usually are obtained with iterative procedure, or more importantly with engineering intuition. By having relatively good initial conditions, implementing the simplex algorithm converge to very satisfying results.

Chapter 5

Verification and Proof of Concept

Within this chapter first the numerical code will be verified according to existing ones, in EXCEL and the MATLAB based EIM Framework done by Olney in [26]. Numerous verification examples were made; however due to brevity, here will be presented only some. The second part of the chapter is devoted to the structural optimization, where first examples will be presented on cross section level, solved by both methods, the nested optimization and with application of equilibrium equations in the constrains, and finally problems on element level are included. The previously cited authors have done numerous tests which are compared to its analytical solution for the general usage of the EIM.

5.1 Numerical verification

5.1.1 Cross section model

The first test is a simple rectangular reinforced concrete section with a tension stiffening zone, modeled in the tension zone within the concrete constitutive law, done in EIM Framework and in an EXCEL code by Olney in [26], displayed on Figure 5.1 with the material properties in Table 5.1. In the part where compression occurs of the cross section, the tension law in the concrete is brittle. Moment curvature diagram was computed, displayed in Figure 5.2, with a moment increment of 1kNm. Excellent results were obtained with which the model of the cross section is verified. The differences in the region after the cracking moment and before the yielding are due to different solvers and tolerances for the objective function and unknown parameters used. MATLAB has better approximation of the solution due to the more powerful optimization toolbox than EXCEL. The difference between the two MATLAB codes comes from the reason that Olney used a tolerance of 10^{-8} for the objective function, the parameters, as well as the approximation of the zero-curvature and strain within the cross section calculation, whilst in this work a tolerance of 10^{-12} is used. Generally, the three results are correct, which is proved by the two minimums within the yielding region due to the tension stiffening material law.

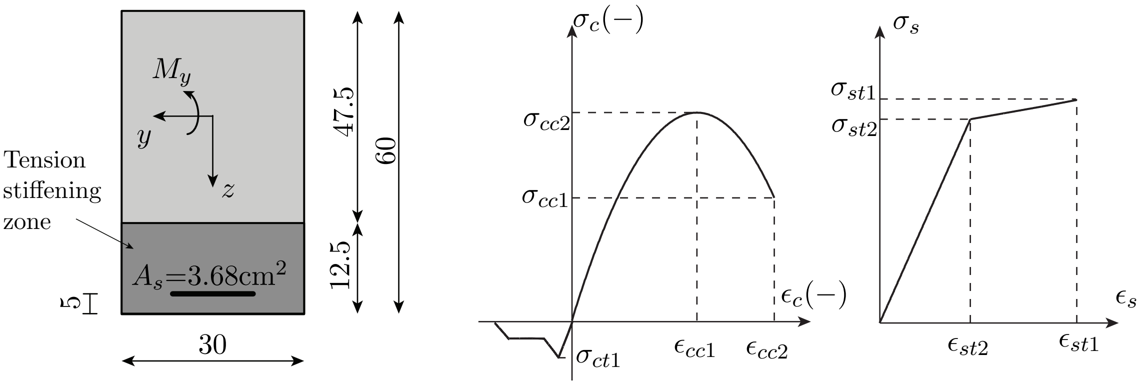


Figure 5.1: Cross section model verification: cross section (left) (units: [cm]); concrete (middle) and reinforcing steel (right) material law, which is bilinear for compression and tension.

Concrete		
	ϵ[‰]	σ[kN/cm^2]
c1	-2.16	-3.80
c2	-3.50	-2.25
t1	0.10	0.34
t2	1.57	0.14
t3	1.85	0.14
t4	7.58	0.00

Reinforcing steel		
	ϵ[‰]	σ[kN/cm^2]
c1	-2.50	-50.00
c2	-25.00	-52.50
t1	1.89	37.87
t2	7.58	52.50

Table 5.1: Cross section model verification: material properties.

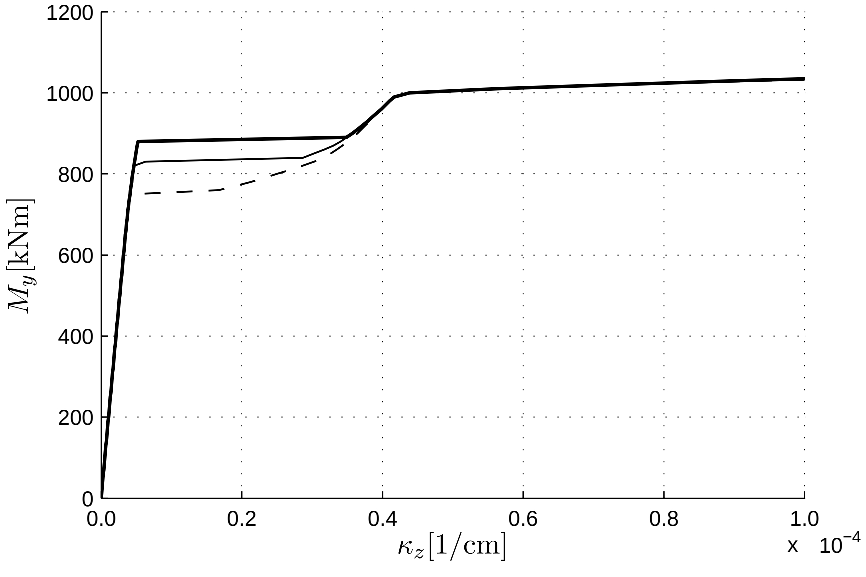


Figure 5.2: Moment curvature diagram for cross section model verification with EXCEL(– –); EIM Framework(—) by Olney in [26] and this work(—). Excellent correspondence; differences in the sensitive region after cracking and before yielding due to different solvers (MATLAB and EXCEL) and different tolerances 10^{-8} (Olney) and 10^{-12} (this work).

5.1.2 Element model

The verification of the element model is done on a 10m cantilever beam, done by Olney in [26], subjected to three forces in different directions. To have a diversity and present different aspects of the code, a composite cross section, along with the parabolic rectangular concrete material law by the EC, disregarding the tension part. The chosen concrete class is $C30/37$ with $S500$ as reinforcing steel and $S235$ for the I profile. System was discretized on five finite elements, using the 3^{rd} order Lagrange and 2^{nd} order Hermite polynomials which corresponds to the 2Node element implemented in the code of this work with four integration points for the internal energy. On Figure 5.4, the results of the displacements, rotation and internal forces are compared. The correspondence is excellent with maximum difference less than 1%.

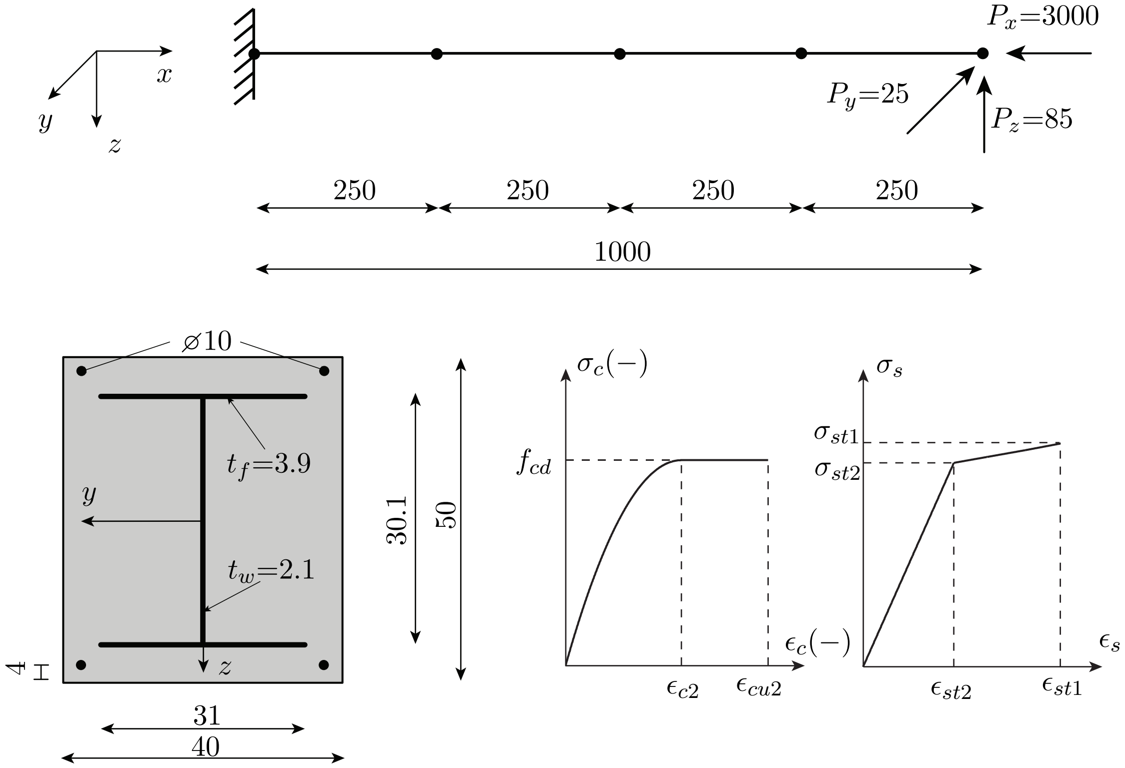


Figure 5.3: Element model verification: element discretization (top); cross section (bottom left); concrete (bottom middle) and bilinear material law (bottom right), which is bilinear for compression and tension (units: [cm]; [kN]).

Concrete			Reinforcing steel			Steel		
	ϵ[‰]	σ[kN/cm^2]		ϵ[‰]	σ[kN/cm^2]		ϵ[‰]	σ[kN/cm^2]
c2	-2.00	-1.98	c1	-2.17	-43.48	c1	-1.19	-23.50
cu2	-3.50	-1.98	c2	-25.00	-45.65	c2	-25.00	-23.50
			t1	2.17	43.48	t1	1.19	23.50
			t2	25.00	-45.65	t2	25.00	23.50

Table 5.2: Element model verification: material properties.

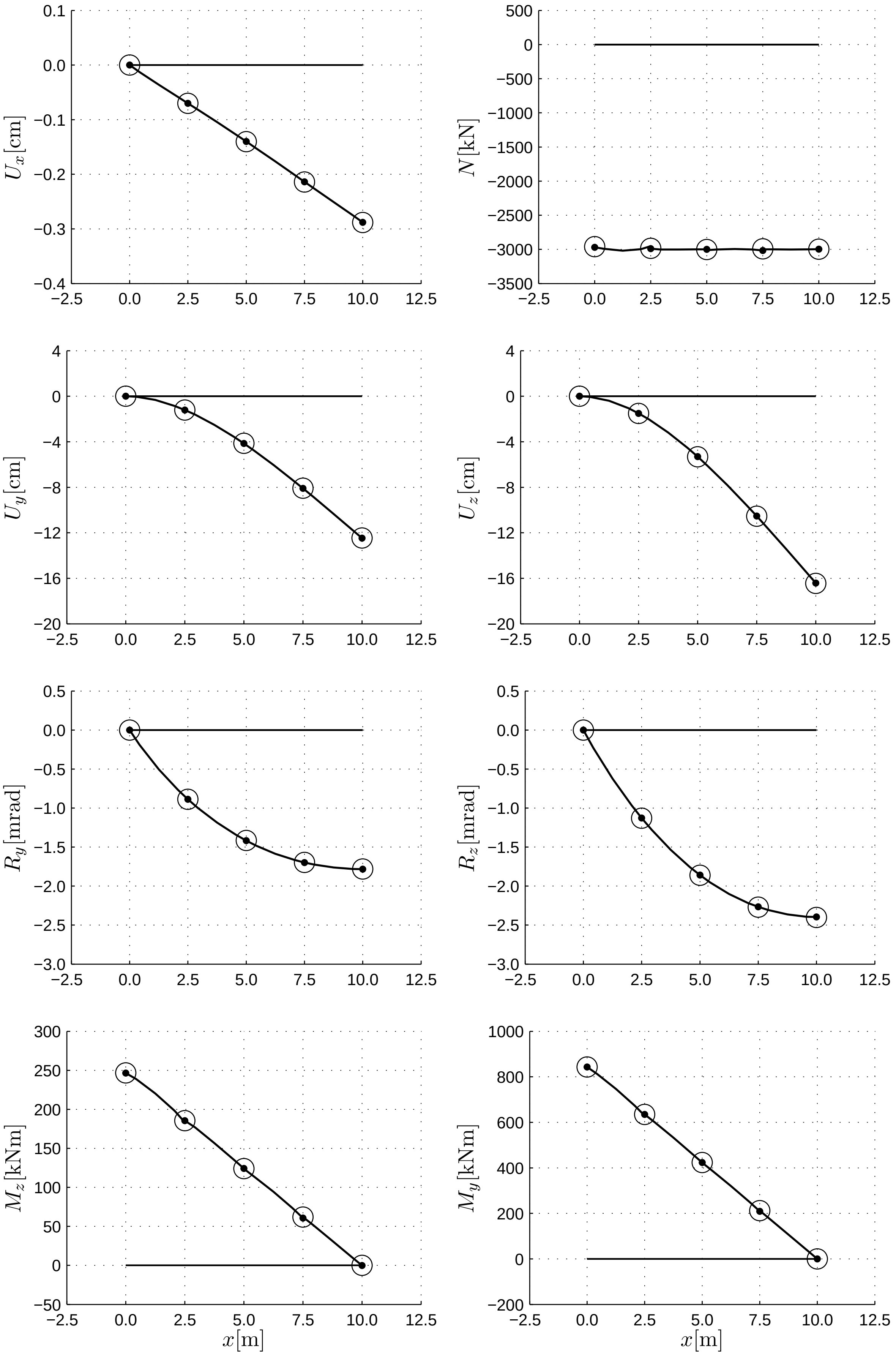


Figure 5.4: Displacements and internal forces for element model verification: this work (-●-) and EIM Framework (-⊖-). Maximum absolute difference is less than 1%.

5.2 Selected structural optimization examples

Having the code validated, multiple structural optimization examples were computed and the majority have performed well. Mainly the parabolic rectangular curve and the bilinear representation of the steel constitutive according to the Eurocode is used due to the inflection point in the nonlinear curve of the concrete constitutive law as discussed before. All of the properties with respect to the constitutive law and material safety factors are documented in the EC [1], here they are only labeled with respect to the ultimate strength. The SO problems are solved with the principle introduced in equation (4.27) and (4.22) and for clarity the problems will be formulated in tabular form; however, they can be formulated in same manner principally. In order not to be constantly repeated, the Newton's algorithm for minimizing the energy was usually set to have 10^{-12} tolerance on the function with 20 000 maximum function counts and 600 iterations. The simplex on the other hand since it is used only externally, tolerances were set according to the function which was minimized and with up to commonly 300 iterations. In the shape optimization, due to its high computational effort, the algorithm was ran multiple times with less iterations, retaining the results from one and using as initial guess within the next one. With this, higher control of the optimization process was established, which helps if the simplex stagnates on a local minimum.

5.2.1 Cross section optimization examples

T-Section: flange and area of reinforcement

Very common design problem in engineering practice is to compute the depth of the flange of a T-Section and the area of steel in order to have good correspondence between the steel and concrete strains. Analytically, the procedure is usually to choose a certain depth of the flange, perform check of the neutral axis, weather is within the flange or not, and then calculate the reinforcement. The desired design is when minimum concrete area is used to take the compression force and the reinforcement is at least yielding, and for appropriate height, balanced failure is the best option. Here both, the area of concrete and reinforcement are calculated simultaneously and with the assumption that the width of the flange is its effective width. Figure 5.5 gives the cross section properties. The concrete is modeled as $C30/37$ with the parabolic rectangular constitutive law for compression, and the reinforcing steel as $S500$ for both, compression and tension, with the bilinear material law without any with stiffness for the post-yielding strength region. Concrete is assumed to have no tensile strength; however a small inclination is implemented for numerical stability with E=0.05kN/cm^2. The example was calculated using both, the nested optimization and applying the equilibrium conditions within the constrains, with properties of the optimization displayed in Table 5.3.

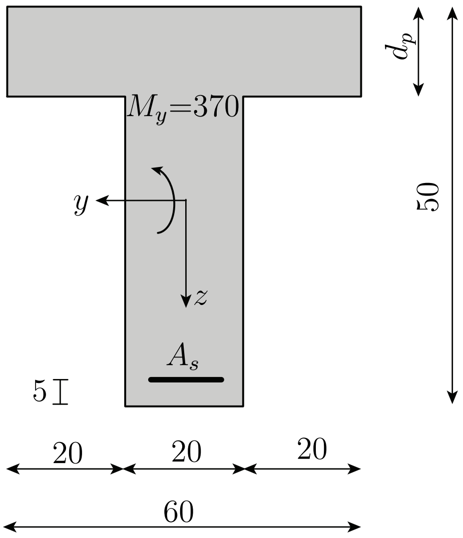


Figure 5.5: T-Section (units: [cm]; [kNm]).

Design parameters	d_p and A_s
Constrains(penalty)	A_s >0cm^2 d_p >0cm $\epsilon_c \geq$-3.50‰ $\epsilon_s \leq$ 25.00‰

	EQ	Nested
Objective ($\rightarrow min$)	$0.9A_s$+$0.1d_p$	$0.9A_s$+$0.1d_p$
Algorithm	interior point	simplex
Initial conditions	d_p=15cm A_s=5cm^2	d_p=15cm A_s=5cm^2

Table 5.3: T-Section: Optimization parameters with explicit equilibrium constrains (EQ) and nested optimization.

The constrained optimization *fmincon* was used in the EQ constrained case with the interior point algorithm due to the convexity of the potential energy function with which the EQ constrains are convex, and MATLAB does not offer the simplex algorithm with constrains; however imposing additional equality constrains resulted with local minimums and without appropriately chosen initial conditions, the result was difficult to obtain and therefore does not correspond fully with the nested, which is be the optimum case; thus it proved to be very sensitive on the initial conditions, whilst the nested optimization was more robust. The simplex algorithm was used in this case, since no convexity can be proven on the objective function, and the constrains were applied as weak penalty functions. The weights were introduced to obtain the minimum area of concrete and steel, satisfying equilibrium conditions. The difference in the results, presented in Table 5.4, is due high sensitivity of the yielding reinforcement, as it was modeled without any slope. On Figure 5.6 the curvature along with the neutral axis is presented, which is just below the flange of the compression zone, proving that the area of the concrete is used to its full capacity. In this case balanced failure was not obtained, due to the limited height of the cross section.

	EQ	Nested	Difference
d_p[cm]	7.72	7.25	4.8%
A_s[cm^2]	19.49	19.09	2.5%
ϵ_c[‰]	-3.50	-3.50	0.0%
ϵ_s[‰]	14.10	14.90	4.7%

Table 5.4: T-Section: Results for the nested and EQ constrained optimization.

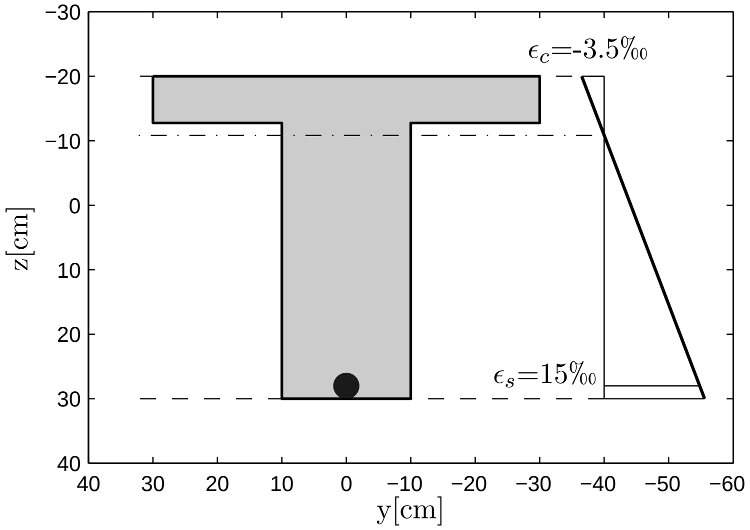


Figure 5.6: T-Section: Computed curvature with the nested optimization. Failure is through compression in the concrete with yielding of the steel. The scale is 1‰=1cm on graph.

Biaxial bending: position of centroid and area of reinforcement

The following example is more of a principle example, in order to test and check the optimization technique concerning as a free coordinate as a parameter, which until some extend, one can relate to the most basic the shape optimization. Rectangular cross section is subjected to biaxial bending from two moments around perpendicular axes, and the goal is to obtain the optimal position of the centroid of the reinforcing steel for minimum area as an objective function. Again, parabolic rectangular concrete compression law is used, class $C30/37$ and bilinear material law for the reinforcing steel, class $S500$, with stifness after the yielding part (f_y=43.5kN/cm^2 and f_u=45.7kN/cm^2). Although this may be perceived as a simple task, there are two possible outcomes: the neutral axis is parallel to the axis above which the moment with the higher intensity is acting, or it crosses both of the axes under certain angle. This depends of the ratio between the moments, i.e. in the first case, there is higher ratio, where the lower moment is balanced with the position and the area, required for the higher moment, of the reinforcement or for the latter case, with balanced ratio between the moments. What describes high or low ratio between the moments depends solely of the cross section properties - $B\backslash H$ ratio in case of rectangular cross section. The choice in this case is the balanced moment ratio. Figure 5.7 and Table 5.5 present the geometrical and optimization parameters. The constrains are taken with respect to the concrete cover of 5cm as well.

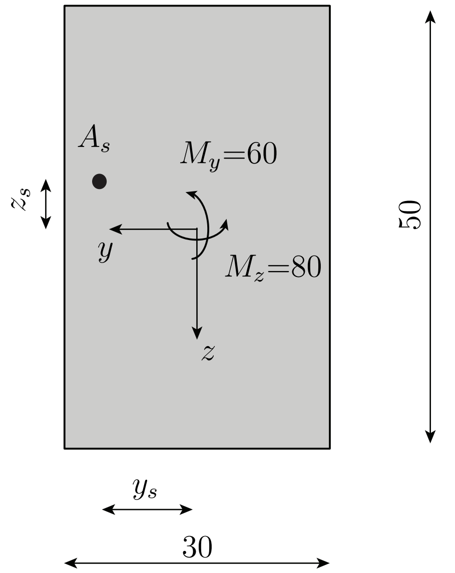


Figure 5.7: Rectanglar section (units: [cm]; [kNm]).

Design parameters		y_s, z_s and A_s
Constrains(penalty)		A_s >0cm^2
		-20cm≥ $z_s \leq$ 20cm
		-10cm≥ $y_s \leq$ 10cm
		$\epsilon_c \geq$-3.50‰
		$\epsilon_s \leq$ 25.00‰
	EQ	Nested
Objective ($\rightarrow min$)	A_s	A_s
Algorithm	interior point	simplex
Initial conditions	y_s=z_s=2cm A_s=1cm^2	y_s=z_s=2cm A_s=2cm^2

Table 5.5: Biaxial bending: Optimization parameters for EQ constrained and nested optimization.

The optimization procedure is generally the same as the one before with a slight change within the objective function: there exists only one variable and therefore a weighting factor was not needed; however a weight was implemented within the penalty functions for the nested optimization in order to have variables of same scale and enforce a stronger penalty. Unlike the previous example, here no difficulties were engaged during the optimization with slight moderation of the initial conditions; however, with a priori from basic engineering practice can confirm that the reinforcement should be in the lower left quadrant i.e. z_s and y_s should be at least positive for the chosen direction of the moments. Great correspondence was achieved between the two optimization methods compared on Table 5.6 and the result of the nested optimization is presented on Figure 5.8.

	EQ	Nested	Difference
y_s[cm]	10.00	10.00	0.0%
z_s[cm]	20.00	20.00	0.0%
A_s[cm^2]	5.93	5.93	0.0%
ϵ_c[‰]	-3.50	-3.50	0.0%
ϵ_s[‰]	11.00	11.00	0.0%

Table 5.6: Biaxial bending: Results for the nested and EQ constrained optimization with excellent correspondence; when checked within the code, the strains in the concrete for the nested optimization were slightly above the design strains due to weak penalty constrains: ϵ_c=-3.50004‰.

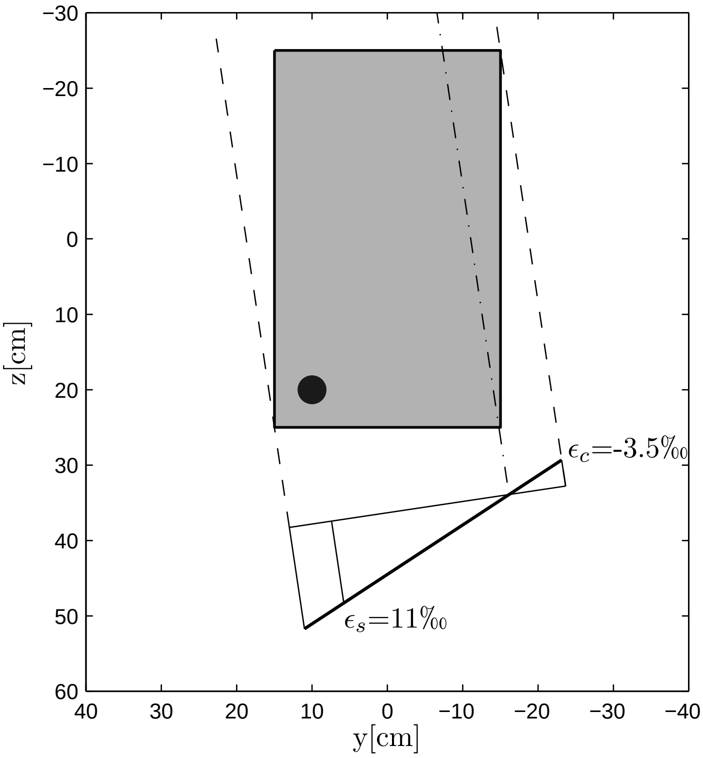


Figure 5.8: Biaxial bending: Computed curvature with the nested optimization. Clear failure through concrete. The scale is 1‰=1cm with respect to the units on the figure.

Composite column: angle of I steel profile and thickness of flange

Composite cross section using steel profiles are commonly used in practice due to their efficiency to withstand high loads with relatively small area. This example presents a complete encasement of steel I profile cross section by concrete with circular form, typical for columns. The problem formulated in this case is for an axial force, to compute the rotated shape of the I section with respect to the bending moment axes, which will result with the least thickness of the flange of the steel profile. Design constrain is the minimum thickness of the flange and the behavioral constrains are according to the material used, in this case, again concrete $C30/37$ and for the steel the $S235$ class, according to the EC, with bilinear constitutive law is used with ultimate strength of f_u=360kN/cm^2. The section, displayed on Figure 5.9, is discretized on 16 equiangular polygon for the calculation in order to describe the circle within the code, and the optimization parameters are displayed on Table 5.7.

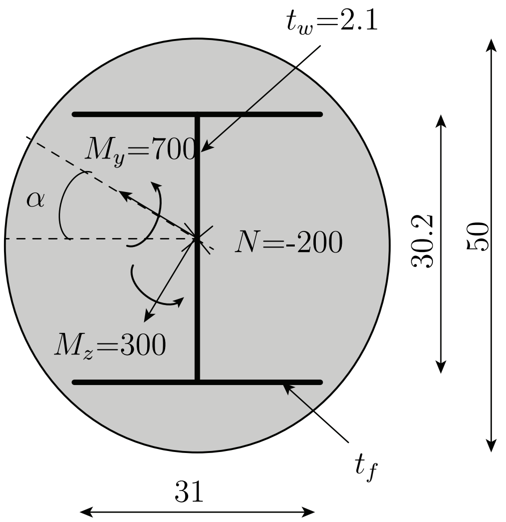


Figure 5.9: Circular section (units: [cm]; [kN]; [kNm]).

Design parameters		t_f and α
Constrains(penalty)		t_f >0.2cm $\epsilon_c \geq$-3.50‰ $\epsilon_s \leq$ 25.00‰
	EQ	Nested
Objective ($\to$ *min*)	t_f	t_f
Algorithm	active set	simplex
Initial conditions	α=0° t_f=1cm	α=0° t_f=1cm

Table 5.7: Composite column: Optimization parameters for EQ constrained and nested optimization.

As expected, the I profile rotated so it its web is perpendicular to the neutral axis. Using both of the methods, well results were obtained; however in this case for the EQ constrained optimization, the active-set algorithm of the *fmincon* function resulted better. This was due to the barrier constrains of the interior point algorithm, which as it was discussed in the previous sections, may not always be exactly on the constrains limit, whilst the active set defines the feasible set and could end up right on the constrain. Again due to weak penalty formulation for the nested optimization, the constrains were violated i.e. the concrete strain were obtained as -3.52‰. An attempt was made to enforce this constrain by employing higher weighting factor, however no better solution was achieved. This could be done manually, by constraining the concrete to a lower strain e.g. -3.4‰, however the results were taken by the "good enough" principle. The results are displayed on the Table 5.8 and Figure 5.10.

	EQ	Nested	Difference
t_f[cm]	2.61	2.62	0.4%
α[°]	23.13	22.93	0.9%
ϵ_c[‰]	-3.50	-3.52	0.5%
ϵ_s[‰]	5.00	5.00	0.0%

Table 5.8: Circular composite column: Results for the nested and EQ constrained optimization with excellent correspondence; again, due to weak constrain, the concrete strain violates the constrain for a small value; however this could be suppressed by manually changing the strain limit of concrete to a lower value.

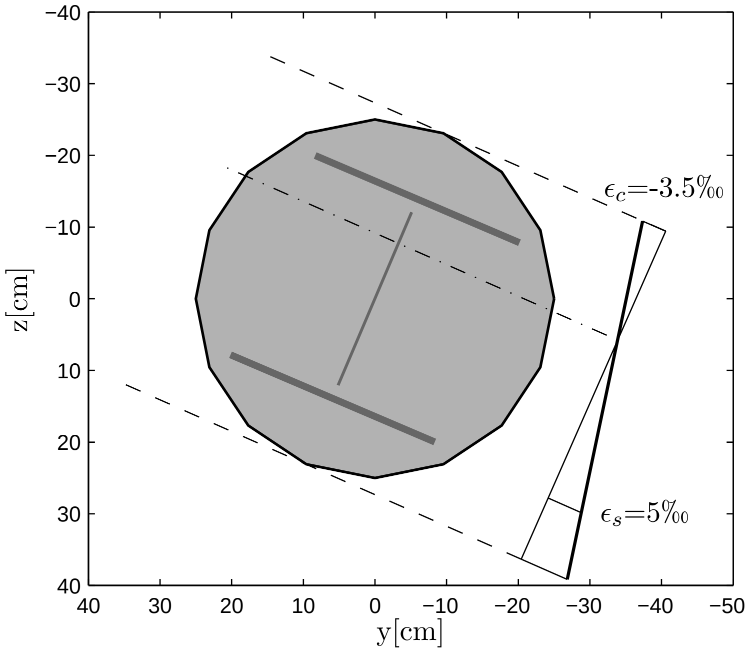


Figure 5.10: Composite column: Computed curvature with the equilibrium constrained optimization. The neutral axis is perpendicular to the web, which brings the conclusion that an optimal position of the profile is achieved. The thickness of the flanges is assumed to be equal on both ends. The scale is 1‰=1cm with respect to the units on the figure.

Elliptical confined section: shape and area

The next example is a composite elliptical cross section, where the steel plates are confining the concrete. These sections are fairly appropriate also for columns due to the confinement concrete which increases its compressive strength and the creep and shrinkage negative effect is fairly decreased; however shear deformation or 3-axial condition is not included. To get closer to the shape optimization in this case an elliptical shape is assumed with the two parameters defining the ellipse a and b. The goal is for a load combination to define the balanced shape, in order to achieve minimum area of the concrete and steel. The balanced area of concrete does not guarantee minimum area of steel due to the fact that the circumference of an ellipse can alter for the same area. Thus, a weighting parameters were introduced. The concrete and steel have the same properties as the previous example, $C30/37$ and $S235$ according to the EC. Figure 5.11 gives the schematic presentation of the system; for numerical calculation the area is discretized on 20 equidistant with respect to the y axis points. The area of the concrete was calculated as the area of an ellipse $A_c = \pi ab$ and the steel area A_s as a sum of the distance between each discretization point, multiplied by a defined thickness t_s=2cm. The parameters were calculated for two different load combination in order to realized the adjustment of the form with the change of the moment ratio and the design and behavioral constrains are contained in Table 5.9 along with the optimization parameters. In this case a weighting factor between the concrete and the steel was assumed as 10 due to the higher cost of the steel with which a Pareto optimal point can be selected. Generally, this is a multi-objective optimization with multiple solutions - infinite solutions. The smallest circumference of a ellipse is a circle, and even in case with higher area, the value of the perimeter might be less.

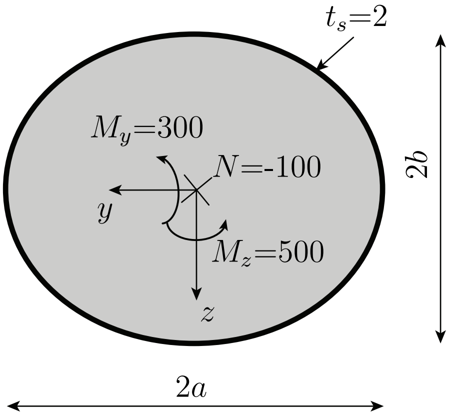


Figure 5.11: Elliptical section properties (units: [cm]; [kN]; [kNm]).

Design parameters		a and b
Constrains(penalty)		0.25≤ $a\backslash b$ ≤4 $\epsilon_c \geq$ -3.50‰ $\epsilon_s \leq$ 25.00‰
	EQ	Nested
Objective ($\rightarrow min$)	$A_c + 10A_s$	$A_c + 10A_s$
Algorithm	interior point	simplex
Initial conditions	a=10(5)cm b=10(5)cm	a=10(10)cm b=10(10)cm

Table 5.9: Elliptical confined section: Optimization parameters for EQ constrained and nested optimization. Initial values in brackets are for the second set of load.

The optimization parameters were chosen as the first examples. In case of the EQ constrained optimization using the interior point algorithm, starting with lower initial values, generally it does not make much difference due to the barrier constrains, with which recovery for function values above constrains is fast. With the simplex algorithm this is not the case, therefore a overestimation of the solution is always better in case of EIM, due to the function values of the energy, and with that of the strains for perfectly plastic material area high. Difficulties were experienced using the EQ constrained method in this case since there is no strict minimum, but more local ones as a consequence of the multi-objective optimization; however using the simplex, this is not a issue. Therefore the concrete area is higher in the EQ case, however if the objective functions are compared, the results differ insignificantly. In the second case, the results correspond well, as it displayed in Table 5.10. On Figure 5.12, the curvature of the first case is displayed. The width is slightly more than two times longer due to the moment ratio of 1.6. The curvature of the second case is displayed on Figure 5.13. Despite the moment ratio of one, the parameters a and b are not the same i.e. it is not a circular shape, as a consequence of the absence of tensile strength in the concrete.

[kN]; [kNm]	N=-100; M_y=130; M_z=200			N=0; M_y=300; M_z=300		
	EQ	Nested	Difference	EQ	Nested	Difference
a[cm]	16.42	16.69	1.6%	14.15	14.11	0.3%
b[cm]	8.41	8.16	3.0%	15.06	15.04	0.0%
ϵ_c[‰]	-3.50	-3.50	0.0%	-3.50	-3.50	0.0%
ϵ_t[‰]	3.86	3.85	0.3%	4.40	4.40	0.0%
A_s[cm^2]	158.04	158.62	0.4%	181.78	181.78	0.0%
A_c[cm^2]	433.88	427.57	1.5%	666.99	666.99	0.0%
$O.F$[cm^2]	2014.28	2013.77	0.0%	2517.79	2517.79	0.0%

Table 5.10: Results of two different load cases for the elliptical confined section. Well correspondence for both: the objective function (O.F) is nearly the same in the both cases; the difference in A_c is due to local minimums of multi objective optimization. Scale is 1‰=1cm with respect to the units on the figure.

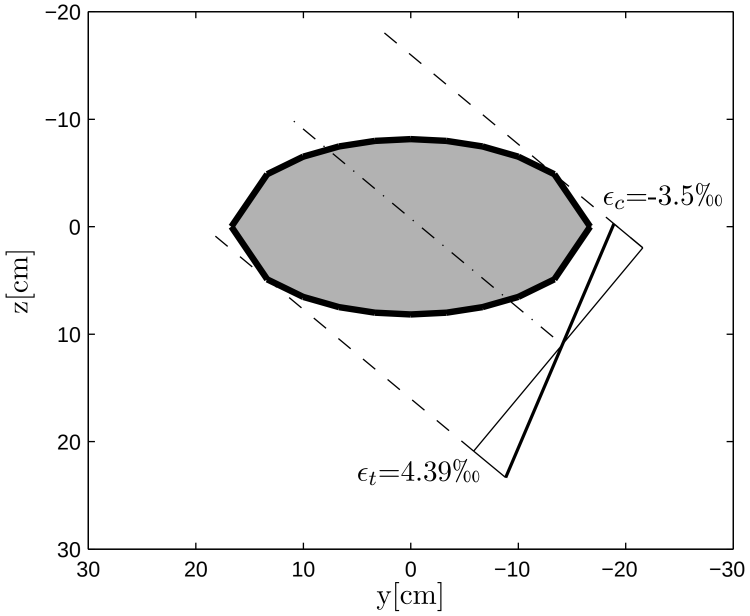


Figure 5.12: Elliptical column case 1: Computed curvature with the equilibrium constrained optimization for N=-100kN; M_y=130kNm; M_z=200kNm. The neutral axis passes through figure on almost two identical surfaces. The width is slightly more than two times longer than the height, due to the moment ratio combined with no tensile force of the concrete, brings the conclusion that the shape is optimized to produce largest moment of inertia. The scale is 1‰=1cm with respect to the units on the figure.

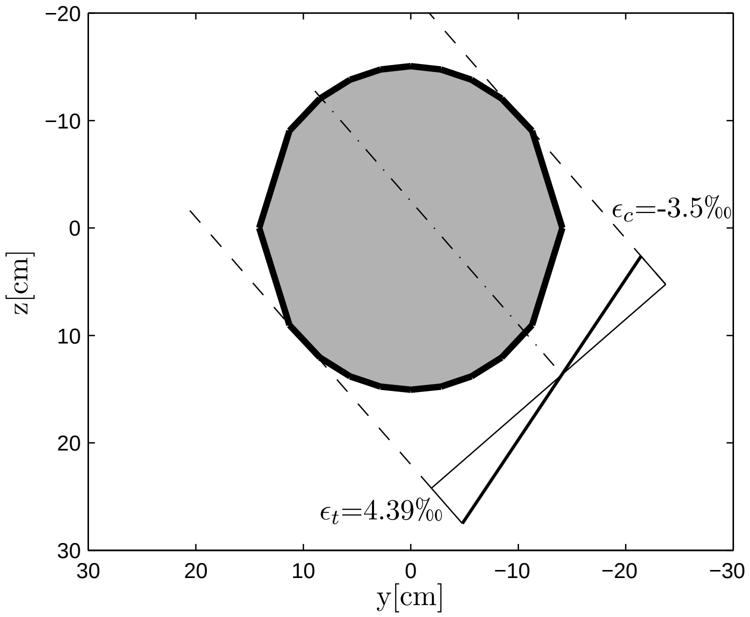


Figure 5.13: Elliptical column case 2: Computed curvature with the nested constrained optimization N=0kN; M_y=300kNm; M_z=300kNm. The ratio between the moments is exactly 1, however the two parameters of the ellipse are not the same, since the concrete is assumed without any tensile strength. The scale is 1‰=1cm with respect to the units on the figure.

5.2.2 Element optimization examples

Structural optimization on element level with the EIM, for now is limited to beam elements. Here, unlike on cross section level, the optimization with equilibrium equations in the constrains can not be used, therefore only the nested one will be introduced. Two types of problems are reviewed: sizing problems, by optimizing a cross section property taking into account the moment distribution, and discrete shape optimization by calculating the position of a discrete node.

I-Steel simply supported beam: height along the length

The first example is a 10m simply supported steel beam with I steel cross section, for which the height over the length is an unknown parameter, for a distributed load. Although this is a basic example, for which the solution can be calculated analytically, it serves as a practice for validation of the method. The material law used is bilinear for steel $S235$ according to the Eurocode, without any stiffness after the yielding point in order to be checked by analytical calculation. As displayed on Figure 5.14,the beam is discretized on 6 elements, each using 5 integration points for both, the internal and external potential energy. Due to symmetry, 3 different cross section have been assigned. On the middle of the span, the two elements have shorter length, since this is the point where the yielding is expected first. The type of the finite element used is the previously described 3Node one.

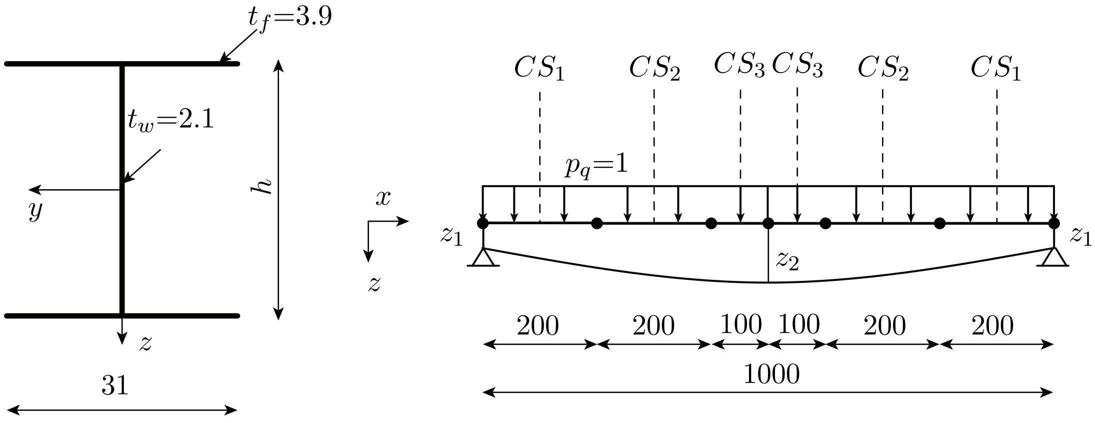


Figure 5.14: I profile height optimization: steel profile cross section with height as unknown dimension (left)(units: [cm]). Discretized simply supported beam on 6 elements (right) Design parameters: z_1, z_2; height h_i at CS_i with Lagrangian polynomial interpolation. (units: [cm] and [kN/cm]).

The idea for structural optimization used in this case is to interpolate the section height with a Lagrange 3^{rd} order polynomial and lower the number of unknowns from 3 to 2, therefore the design parameters are the values $[z_1, z_2]$ and the end and middle of span, although from engineering practice, it is known that z_1=0cm, since no moments exists in the support, it was assumed in this case to test the optimization method. As an objective function, due its continuity and good performance, the one presented in (4.26) is used, where the strains are maximized, further referenced as $C(\epsilon, U)$. Design constrain of 15cm for the height was assumed as a weak penalty function, and with initial conditions, which were expected to be higher than the optimal ones and the behavioral constrains are according to the material properties. Using this a priori engineering knowledge, the simplex algorithm was employed. Summarized, the optimization parameters are presented on the following table:

Design parameters	a_1 and a_2
Constrains(penalty)	h >15cm
	$\|\epsilon_s\| \leq 25.00‰$
Objective ($\rightarrow min$)	$C(\epsilon, U) + z_1 + z_2$
Algorithm	simplex
Initial conditions	z_1=5cm, z_2=40cm

Table 5.11: I profile height optimization parameters.

The resulting heights of the optimization are presented in Table 5.12, along with the strains in the cross section of the plastic hinge, in the middle of the span displayed on Figure 5.15. The results showed excellent correspondence with the analytical method of computing the plastic moment M_{pl}^{cal} for the mid-span section, using the plastic section modulus which is calculated with the assumption there is full plastification in the section, whilst in the optimization is clear that the curvature is limited; thus the slight difference appear. On Figure 5.16 the change of the cross section is plotted against the length and compared with the theoretical quadratic parabola for continuous load, and as it can be concluded the correspondence is well. On Figure 5.17 the moment diagram is shown for the beam with optimized cross sections. The discontinuities due to the yielding of one of the cross sections at the point where the height change occurs.

h_1	h_2	h_3	ϵ_s	M_{el}	M_{pl}^{cal}	Difference
[cm]	[cm]	[cm]	[‰]	[kNm]	[kNm]	[%]
24.73	37.09	38.64	25.00	1249.85	1272.17	1.75

Table 5.12: I profile height results: h_i between centroid of flanges at each CS_i. M_{el} is elastic moment at span, M_{pl}^{cal} is calculated analytically plastic moment from the section properties at span. The difference is due to the assumption of infinite curvature calculating the plastic moment, whilst in this case the curvature is limited; overall it is excellent correspondence.

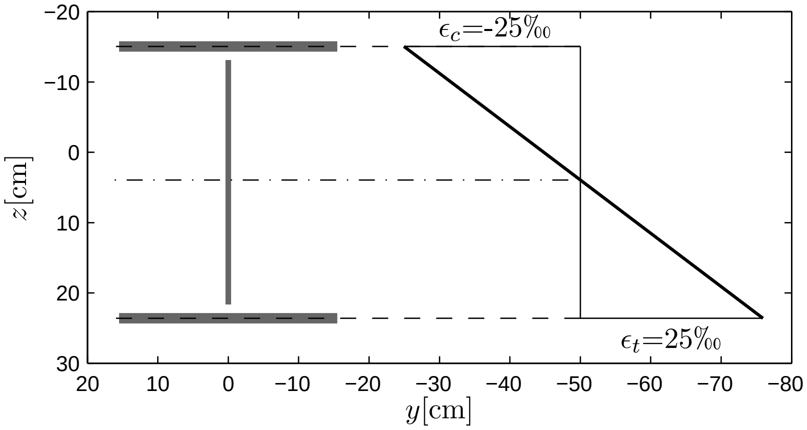


Figure 5.15: The critical section in span; expected balanced failure. The scale is 1‰=1cm with respect to the units on the figure.

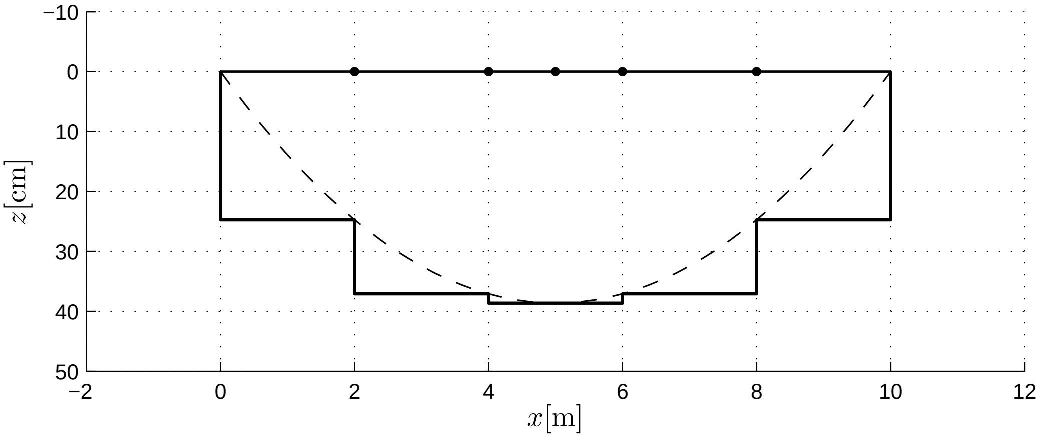


Figure 5.16: I profile height over length: height of cross section (—) intersects perfectly the points of the theoretical height for distributed load, the quadratic parabola (– –).

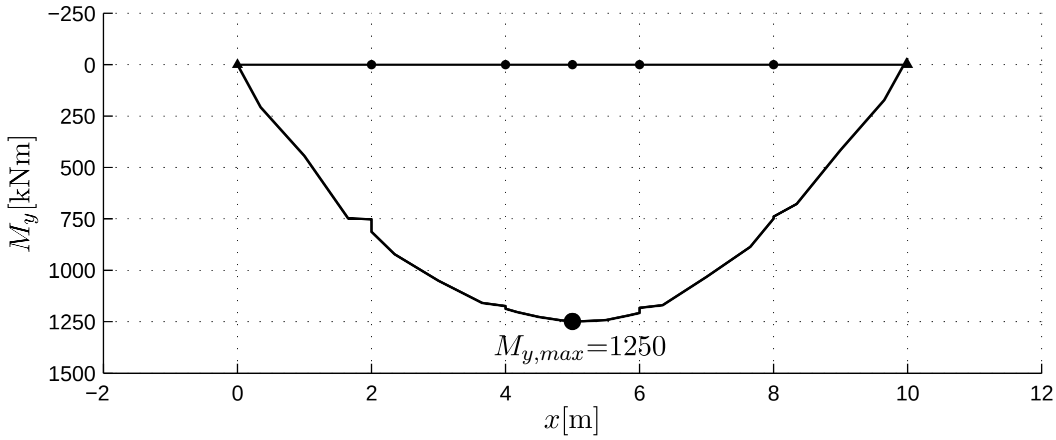


Figure 5.17: I profile height bending moment diagram: the discontinuities that appear on the diagram where different section meets are since the end section is yielding in one of the finite elements and in the other not, as it was optimized.

Two span composite girder: area of reinforcement at support

The next example from sizing problems is a 50m two-span girder, with a composite cross section of I steel profile and reinforcement slab on top, subjected to dead and live load, for which the amount of reinforcement in the span has to be optimized taking into account the moment redistribution. At first, the height of the I profile is optimized using the cross section optimization, with the worst combination of live loads i.e. only one of the spans is subjected to a live load. The idea is so have the same cross section along the length, and only add required reinforcement at the support cross section to withstand the support moment, which is the highest if both spans are subjected to live load. Commonly, within any legislation the redistribution of moments is taken into account as a certain percentage. Here the goal is, assuming the rotational capacity of the hinges satisfy, a comparison of the required reinforcement computed with plastic design between the elastic and redistributed moments to be made. The system is displayed on Figure 5.18 along with the load and its discretization. For discretization, the 3Node element which means it adds additional node for every one of the 8 finite elements used. The system is discretized in this way in order to have dense meshing near the support. Five integration points are selected for the internal and external energy. The cross section with the additional reinforcement is in the 4 elements near the support. The parabolic rectangular material law for concrete $C40/50$ according to the Eurocode is used. The steel for the I profile is $S235$, and $S500$ for the reinforcement, both including stiffening after the yielding point, using the bilinear material law.

On Figure 5.19 the optimized height of the cross section in span for elastic moments of $LC2$ is displayed. The procedure was done using the cross section optimization code, by maximizing the strains.

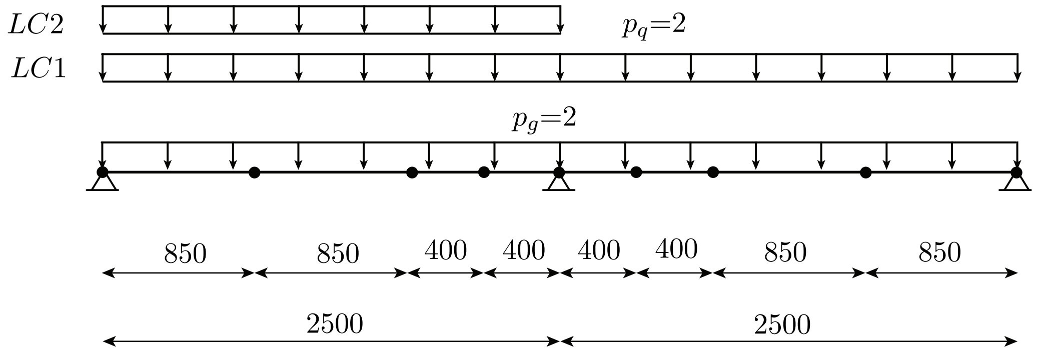


Figure 5.18: Composite beam reinforcement optimization: discretization of the system; two load cases applied, $LC2$ for computing the height of cross section level in span and $LC2$ for computing the additional reinforcement in the support taking into account moment distribution (units: [cm] and [kN/cm]).

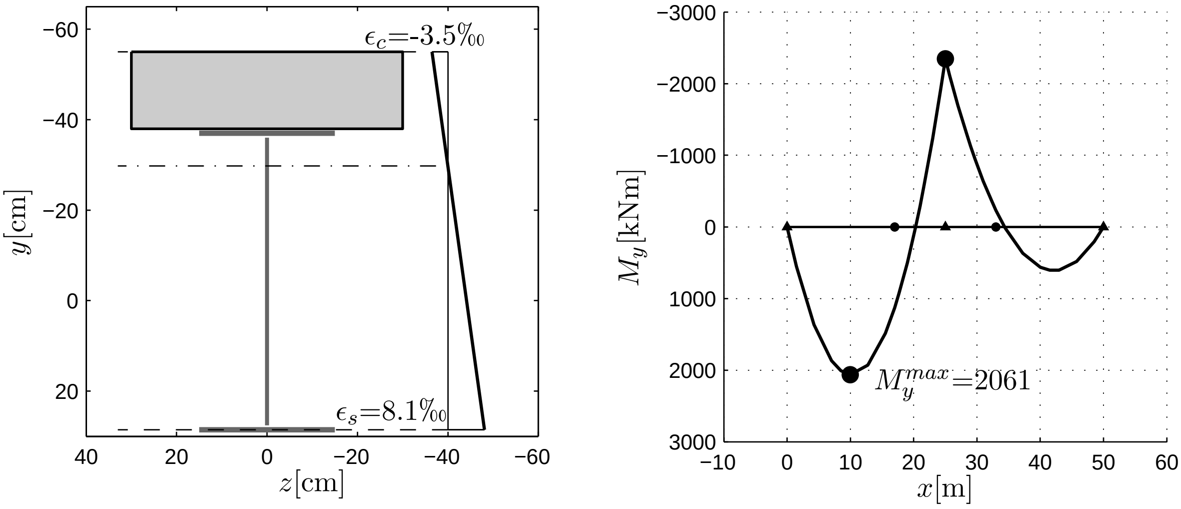


Figure 5.19: Optimized height of I steel profile for cross section in span for elastic moment from load case 2 ($LC2$). Strain distribution: compression failure (left), elastic bending moment moment M_y (right).

Table 5.13 display the optimization parameters used for the optimization of the reinforcement in the support. This was done using the $C(\epsilon, U)$ function as an objective function, since there is only one design parameter, and therefore the optimum would be when the strains are maximized in any section along the length. For sizing problems, as proven before simplex algorithm is a good choice, especially for less parameters, as in this case, there is only one. On Figure 5.20 the support cross section is given, which has slight increment in the height from the previously optimized one in order to have rounded parameters. The height of the cross section is uniform along the length.

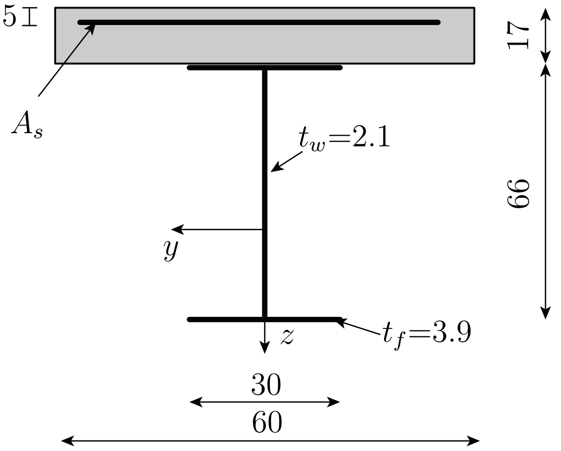


Figure 5.20: Modified support composite section (units: [cm]).

Design parameters	A_s
Constrains(penalty)	A_s >0cm^2
	$\|\epsilon_s\| \leq$ 25.00‰
	$\epsilon_c \geq$ -3.50‰
Objective ($\rightarrow min$)	$C(\epsilon, U)$
Algorithm	simplex
Initial conditions	A_s=30cm^2

Table 5.13: Composite girder: Optimization parameters.

The results of the optimization are presented in Table 5.14. As displayed on Figure 5.21, the reduction of the moment is 35% in support due to the redistribution of internal forces. The required reinforcement was also calculated for the elastic moment with the cross section calculator, and if the full distribution is used with the assumption that the rotational capacity satisfies, it leads to a 70% reduction. In practice, this would not occur as the rotational capacity would not be satisfied. The optimization procedure was done step by step i.e. setting less iteration and running the optimizer more than once with the solution obtained from the previous step as initial conditions, in order the results to be constantly monitored. Figure 5.22 illustrate the distribution of the elastic and plastic displacements. Due to the redistribution of forces and plastification of the standard and reinforcing steel, the maximum plastic is two times larger than the maximum elastic displacement. The failure occurs in the support through failure in the reinforcement on the top with 25‰, whilst the strains in the bottom flange are 24‰. Typically, failure would occur through buckling or shear, however this is not considered, along without any stiffeners in this case. The curvature at failure is displayed on Figure 5.23.

Critcal CS	ϵ_t	ϵ_c	A_s^d	A_s^{el}	Difference
	[‰]	[‰]	[cm^2]	[cm^2]	[%]
Support	25.00	24.00	13.26	51.19	74.00

Table 5.14: Composite girder results: ϵ at tension and compression. A_s^d is the area of steel computed taking into account redistribution at support section, whilst A_s^{el} is computed without taking into account the redistribution.

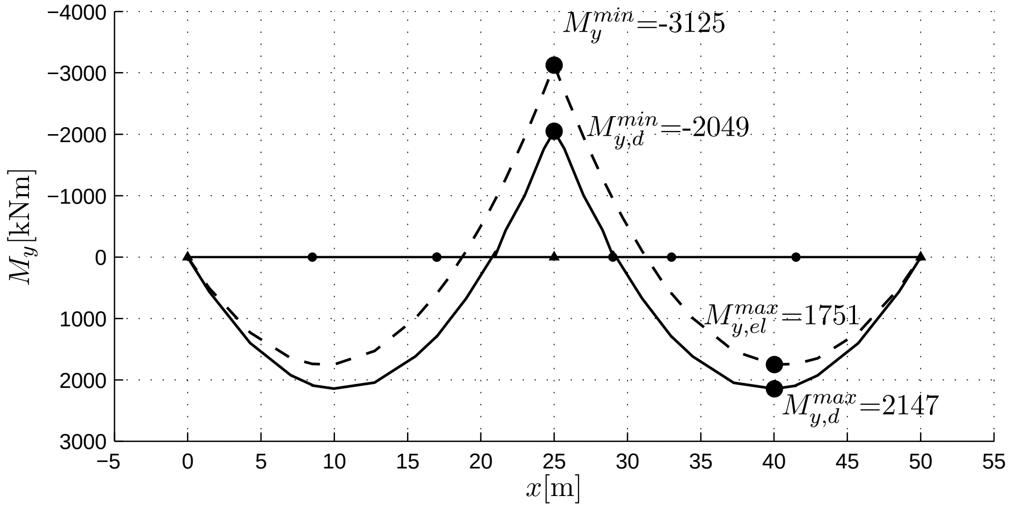


Figure 5.21: Composite girder: Comparison of the elastic $M_{y,el}$ (– –) and distributed bending moments $M_{y,d}$(—–) for $LC1$. The reduction of the elastic moment in support is 34.57% and in span, an increment of the positive bending moment of 17.44% consequently. Due to the increment of the height of the cross section for design reasons, the failure did not occur in span even though the moment of distribution is higher then the elastic for $LC2$; however the cross section is yielding.

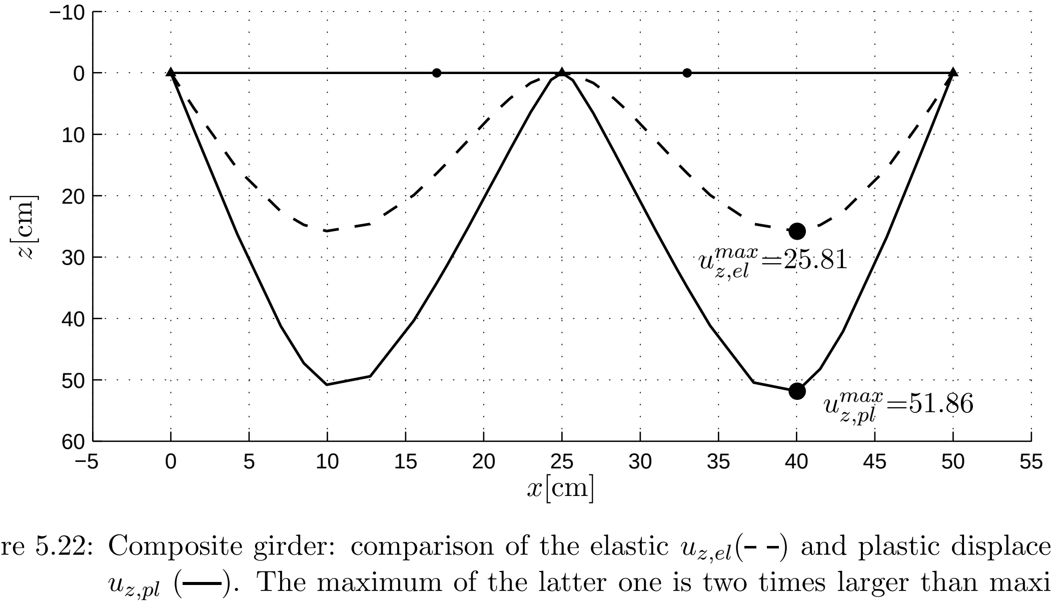


Figure 5.22: Composite girder: comparison of the elastic $u_{z,el}$(– –) and plastic displacement $u_{z,pl}$ (—–). The maximum of the latter one is two times larger than maximum of the former one.

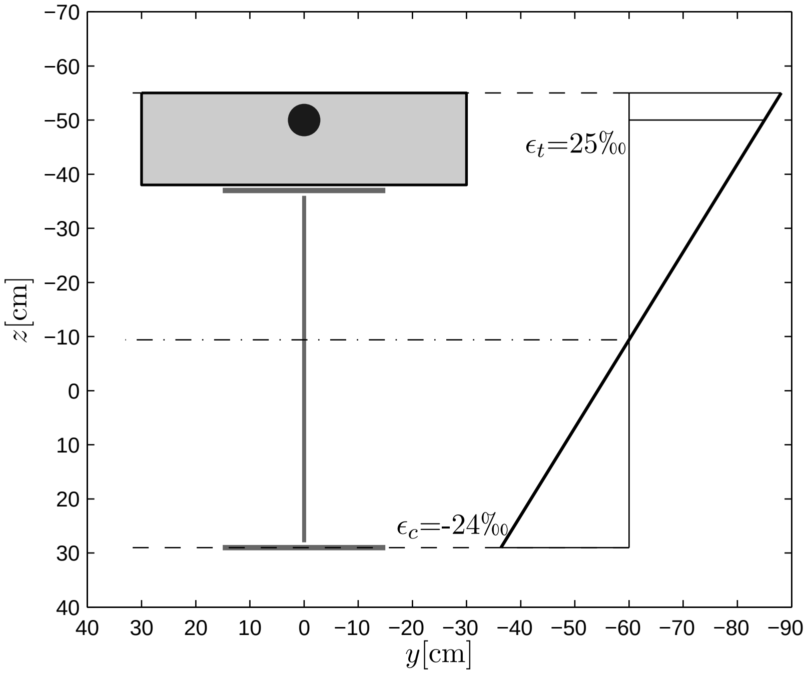


Figure 5.23: The critical section at support, where failure occurs in the tension at the top reinforcement. Practically, the failure would be through buckling or shear in the bottom flange; however here, since it is an academic example, this is not considered. The scale is 1‰=1cm with respect to the units on the figure.

Arch bridge: thrust line

The thrust line concept presents a shape of a structure, an idealized curve for which only axial forces would appear. The shape depends solely on the loading condition. The goal within this principle example is to compute this shape for different loading situations by modifying the coordinates of the discretized system for a set of design constrains. Here it is done for one system with two different loading combination including the two types of loads: discrete and continuous. This example may be categorized in the discrete shape optimization category since the design parameters are the z coordinates of the nodes. The system is a 12m span girder for which as a design constrain the node in the middle is set to be on a 2m height. Although this is relatively short span, the inspiration comes from the arch bridges where the rise-to-span ratio $f\backslash L$ is a preset dimension. As a cross section the I profile, computed for the circular composite column example in the previous section is used, and since it is uniform along the length on a simply supported system , it is trivial to describe it. A linear elastic material is assumed with E=20 000 kNm/cm^2. On Figure 5.24 the discretization is displayed with the two different load cases. The $LC1$ was chosen to be acting as self weight, in order the resulting shape to be compared with catenary. The system is discretized in 6 equidistant elements using the 3Node finite element.

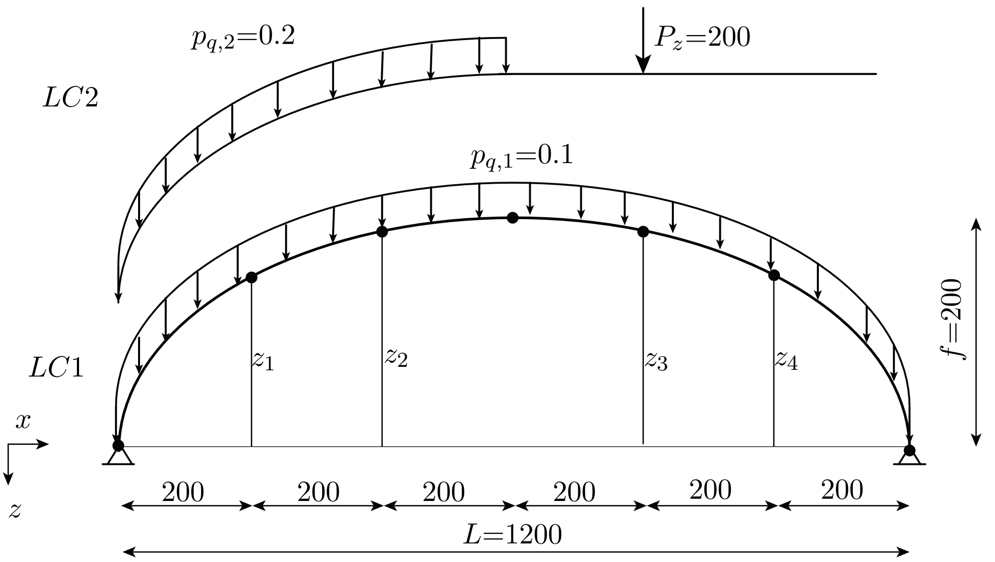


Figure 5.24: Thrust line: discretization of an arch with two load cases, for computing two different thrust lines (units: [cm]; [kN]; [kN/cm]).

For the shape optimization, a hybrid algorithm, has proved to be very efficient, beside its high computational effort. In this case first the genetic algorithm (GA) was used with 50 individuals as initial population and 200 generations. The genetic algorithm, within MATLAB does not offer any initial conditions to be set, only upper and lower boundaries for the unknown parameters; thus it was only set up from 0 to -10m for each unknown to obtain the region where the initial guess is expected. They were all scaled and the initial population was done within the range of [-1,1]. After a result was obtained, since the genetic algorithm is not a deterministic and it is very unlikely to converge to a minimum, the simplex was used to refine the results. The optimization properties are shown on Table 5.15. The objective function in this case, was to minimize the maximum absolute moment i.e. $|max(M)| \rightarrow$ min. For the first load case, due to symmetry there were only two unknown parameters, and in the second four. After obtaining the initial conditions, with iterative procedure using less steps, in order to control the optimization, the results were obtained.

	$LC1$	$LC2$
Design parameters	z_1 and z_2	z_1, z_2, z_3, z_4
Constrains(penalty)	z_i <0cm	z_i <0cm
	z_i >-10m	z_i >-10m
Objective ($\rightarrow min$)	$\|max(M)\| \rightarrow$ min	$\|max(M)\| \rightarrow$ min
Algorithm	hybrid (genetic + simplex)	hybrid (genetic + simplex)

Table 5.15: Thrust line optimization parameters.

The results from the optimization for both cases are presented in Table 5.16. To check whether the results are relevant or not, for the first load case they were compared with the theoretical shape for dead load, the catenary. Equation (5.1) defines the catinery with the lowest point of the sag as coordinate center. In order to compute the relation, the horizontal force was needed, which was calculated using the length of the arch S=1284cm, as it is in the theoretical relation, computed with the optimization which is slightly higher then the span:

$$y(x) = \frac{H}{q}\left(\cosh\left(\frac{qx}{H}\right) - 1\right), \tag{5.1}$$

where q is the distributed loading and H=93.1kN is the horizontal force at the lowest point of the sag (or rise as in this case) f computed as:

$$H = \frac{w}{8f}\left(S^2 - 4f^2\right). \tag{5.2}$$

The insignificant difference, between the theoretical and computed shape for $LC1$, is only in one of the nodes and is due to the discretization of the system on finite elements. Figures 5.25 and 5.26 show the two computed shapes and for the second load case, it is clear the straight part of the shape which is attributed to the concentrated loading. From Figures 5.27 and 5.28, which display the internal forces, as it may be recognized, that the intensity of the bending moments is very small compared to the ones computed for straight beam in the figure caption; thus these are just parasitic moments. The major load bearing mechanism is through compression.

		$LC1$		$LC2$
	Calculated	Catinery	Difference	Calculated
z_1[cm]	-117.47	-112.74	4.3%	-118.18
z_2[cm]	-178.38	-178.44	0.0%	-182.45
z_3[cm]	-178.38	-178.44	0.0%	-198.81
z_4[cm]	-117.47	-112.74	4.3%	-95.52

Table 5.16: Thrust line optimization results.

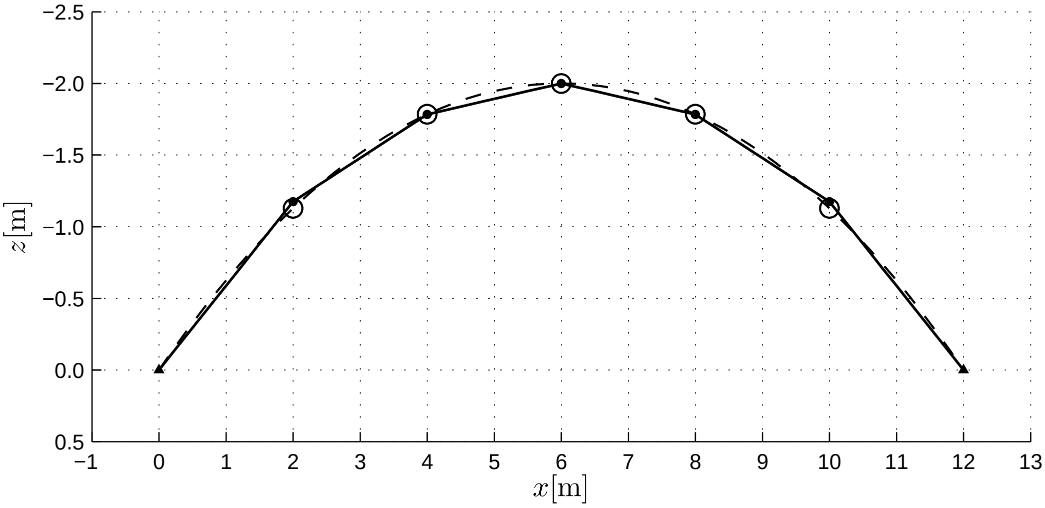


Figure 5.25: Thrust line result for $LC1$: calculated(-●-) and theoretical catinery (-⊖-), excellent correspondence.

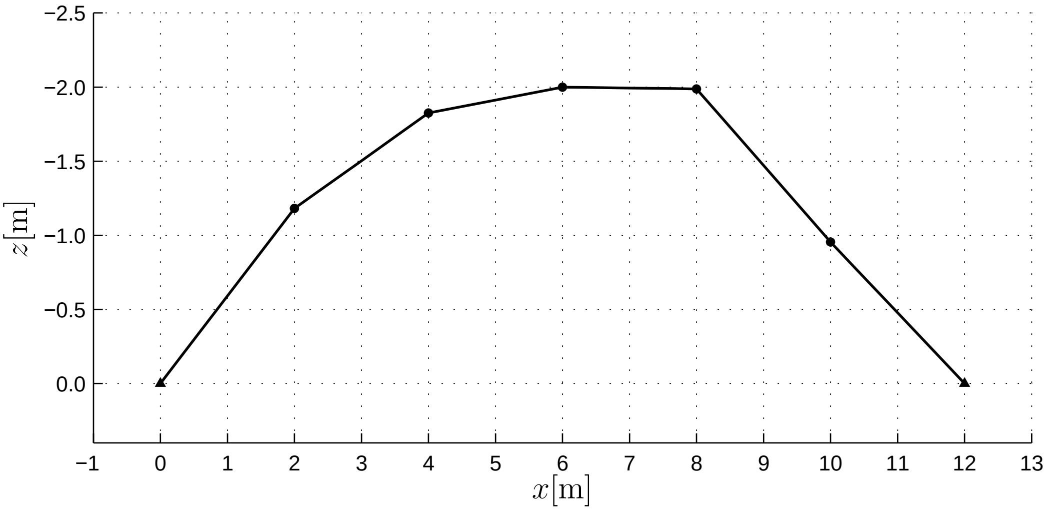


Figure 5.26: Thrust line result for $LC2$: clearly visible where the continuous load is acting, and where the straight part is due to the concentrated force.

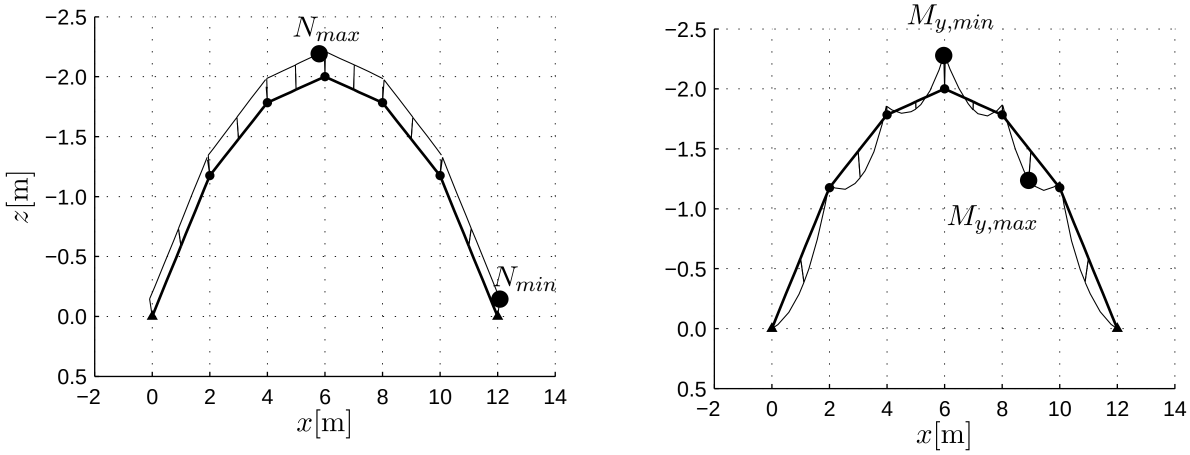


Figure 5.27: Internal forces for $LC1$: Normal forces (N_{max}=-83.27kN and N_{min}=-105.78kN)(left) and residual moments due to discretization ($M_{y,max}$=5.07kNm and $M_{y,min}$=-5.58kN) (right). For comparison, for a straight beam, the maximum moment would be 180kNm.

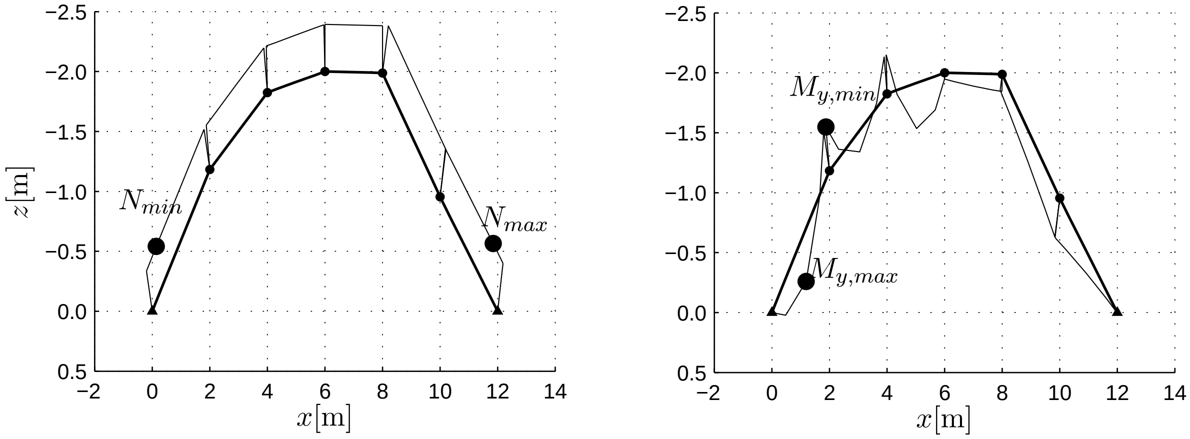


Figure 5.28: Internal forces for $LC2$: Normal forces (N_{max}=-195.24kN and N_{min}=-221.53kN)(left) and residual moments due to discretization ($M_{y,max}$=7.71kNm and $M_{y,min}$=-7.70kN) (right). For comparison, for a straight beam, the maximum moment would be 386.7kNm.

Truss bridge

The last selected problem involves both, sizing and discrete shape optimization. It is based on previous work by Soh & Yang in [49] who compared a solution of a truss bridge with one previously optimized by Saka in [32]. They have used the genetic algorithm and compared several load cases involving a topology optimization for some, however in this case those will be disregarded. Shape optimization problem has been identified as more difficult but more important task than mere sizing problems, since the potential savings in material can be far better improved by the latter [49]. The structure presents a 24m spanned truss bridged, displayed on Figure 5.29, for which the area of the bars are the sizing variables and the along with the coordinates x_6, x_7, z_6, z_7, z_8, resulting in total of 10 variables, taking into account the symmetry.

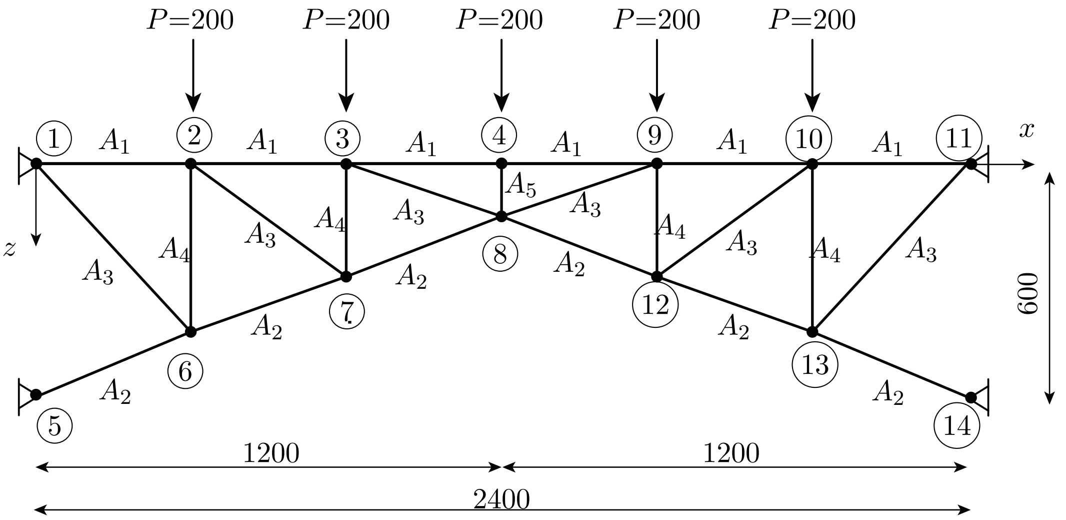


Figure 5.29: Truss bridge disposition. Symmetry is applied, thus the position of nodes 1, 2, 3, 5, 6 and 7 are correspond to the position of nodes 11, 10, 9, 14, 13 and 12 respectively (units: [cm]; [kN]).

For the modeling of the bridge, symmetry is employed, by constraining the x direction in nodes 4 and 8, also dividing the force in node 4. The material is linear elastic with Young's modulus of E=2.1E6 kN/cm^2. Here, a 2Node truss finite element was employed for the bars which uses only the 3^{rd} Lagrange polynomial for longitudinal displacements. The constrains and the optimization parameters are presented in Table 5.17. The objective is the weight of the structure is to be minimized, by minimizing the total volume of steel used for the configuration, by constraining the displacements and stresses. Soh & Yand also include the x_2 and x_3 as design variables without any notice of constrains, which resulted in this work with senseless results as the node is moving to the support, thus the force simply has no influence. In order to compare the results, these variables were taken from the cited authors solution as fixed.

Design parameters	x_6, x_7, z_6, z_7, z_8
	A_1, A_2, A_3, A_4, A_5
Constrains(penalty)	A_i >0.5 cm^2
	u <1cm
	w <5cm
	σ <14 kN/cm^2
Objective ($\rightarrow$ *min*)	$V = \sum_{i=1}^{n_{bars}} L_i A_i$
Algorithm	hybrid (genetic+simplex)

Table 5.17: Truss bridge optimization parameters.

The hybrid method, again in this case for shape optimization, has proved to be the most favorable method. Since the genetic algorithm in MATLAB does not offer the possibility of initial condition, rather boundaries of the variables and its initial range, all of the variables were scaled to have approximate dimension of 0.1 with an preset initial range [-1,1]. The maximum generations were set to 200 with a population size of 20 individuals. The finessing with the simplex was done in more than 20 runs, each using previous solution as initial value, with approximate 200 iterations. In the both of the afore discussed studies, the authors have taken initial conditions, by dividing the structure on equidistant parts, here the lower and upper bounds of each variable are approximately set to have sensible answers with respect to the engineering practice, yet still make broad enough design space to have flexibility. The results from the optimization are presented in Table 5.18, with comparison of the specific weight same as the one for steel i.e. 8000kg/m^3, since it was not specified in the article, and the simple calculation by dividing the weight calculated with the volume resulted in inappropriate dimension. It is assumed that some kind of scaling factors are used. For this calculation, the volume and consequently the weight, of the shape presented the article was computed manually with respect to the given coordinates and the area of the rods.

Variable	Lower bound	Upper bound	Saka (1980)	Soh&Yang (1998)	This work
A_1[mm^2]	0.05	inf	30.00	27.17	9.30
A_2[mm^2]	0.05	inf	5236.00	5136.46	4676.49
A_3[mm^2]	0.05	inf	98.00	106.68	389.84
A_4[mm^2]	0.05	inf	1442.00	1433.24	1460.75
A_5[mm^2]	0.05	inf	1429.00	1420.78	1475.84
x_2[cm]	fixed	fixed	171.00	162.20	162.20
x_3[cm]	fixed	fixed	600.00	579.30	579.30
x_6[cm]	100.00	600.00	195.00	167.30	176.51
z_6[cm]	100.00	600.00	426.00	435.00	433.43
x_7[cm]	300.00	1200.00	599.00	581.20	604.93
z_7[cm]	100.00	600	209.00	184.10	158.56
z_8[cm]	0.0	600.00	100.00	61.40	38.42
Weight [kg]			1303.10	1289.49	1259.23

Table 5.18: Truss bridge results: the reduction of the weight is 3.37% and 2.35% with the results of Saka and Soh&Yang respectively (fixed indicated that the coordinates were taken from the latter, as discussed previously).

Favorable results were obtained, as the weight was reduced by 3.37% with the earlier and 2.35% with the more recent study. Comparison of the shapes are displayed in Figures 5.30 and 5.30. It is clear that the layout conform an arch, as it is more shaped with this approach. The diagonal rods in this case are with higher slope, by which the horizontal component of the force is greater, thus their area is larger. However, due their shorter length than the bottom arch, the weight is decreased. On Figure 5.32 the axial force diagram is presented, with the maximum compression force indicated, which is in the bottom arch as expected.

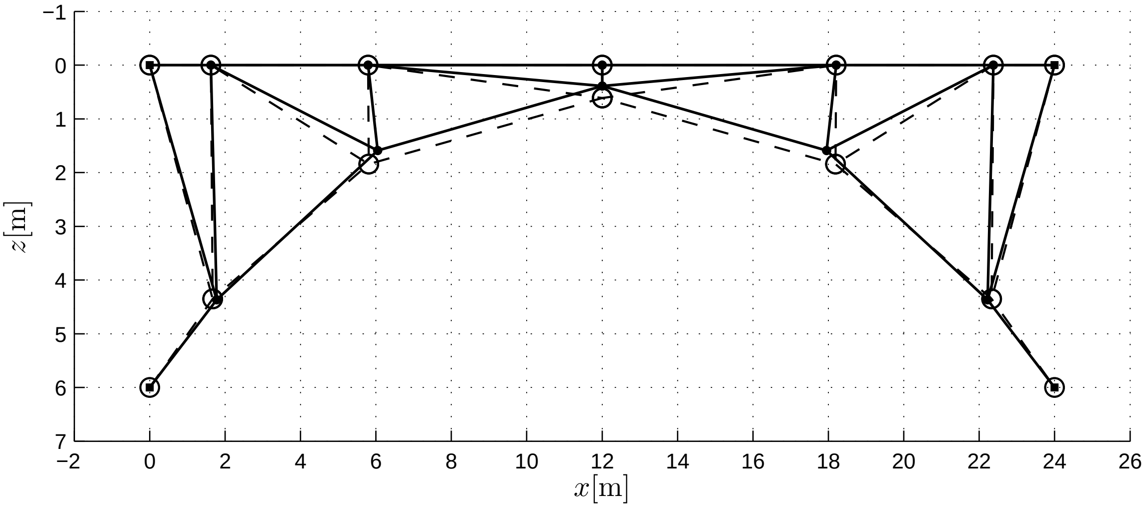


Figure 5.30: Comparison with previous studies: this work (-●-) Soh&Yang (-⊖-). The reduction of weight is due to the reduction of the diameter of the bottom arch rods, and increment of the area of the diagonal members as they are displaced in a position, in which the horizontal component of the force increases. The length of the diagonal members is shorter than the bottom arch; thus less volume is required. The layout conform to an arch.

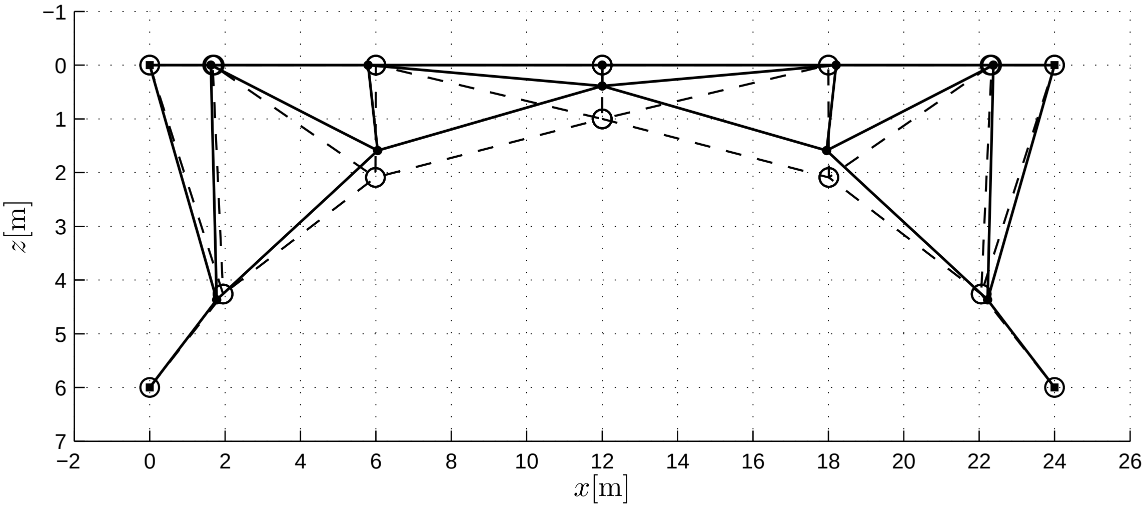


Figure 5.31: Comparison with previous studies: this work (-●-) Saka (-⊖-).

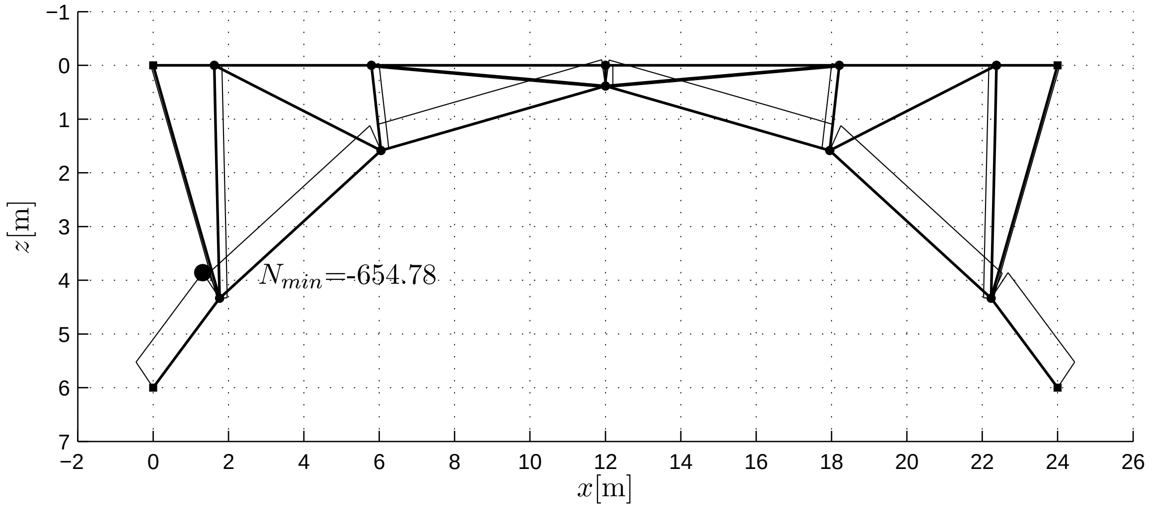


Figure 5.32: Truss bridge: axial force diagram. The scale is 1m=1000kN with respect to the units on the figure.

Chapter 6

Summary and Conclusions

6.1 Summary

The main issue addressed within this work was to research and implement an innovative approach for structural optimization using an alternative method for nonlinear analysis, the Energy Method with Integral description of Material behavior (EIM). The work may be summarized in three parts.

In the first phase the two fundamental areas, the variational formulation on which the EIM is based along with EIM itself and the mathematical optimization are briefly reviewed in the first chapters in order to describe the essence of the structural optimization using the EIM. The differences with the standard Finite Element Method (FEM) are discussed, with which the potential of the EIM may be realized. With respect to the optimization field, firstly the mathematical optimization is reviewed, the analytical solution along with the numerical one for unconstrained and constrained optimization. Each algorithm used within the work was principally presented along with its advantages and disadvantages with respect to the optimization problem. Here, initially the bilevel optimization as a method of the structural optimization with the EIM is proposed, which later has proved successful. At the end of the theoretical part the two fields are brought together as the formulation of the structural optimization problem is made, describing the principles of sizing problems and discrete shape optimization, which are the limitations of the current state of the EIM method.

A numerical MATLAB based code is introduced in the second phase for the nonlinear analysis discretized system, on cross section and on element level. The integral description and numerical difficulties which were encountered of the commonly used constitutive laws were addressed along with different shape functions and integration rules which were used on element level with references for its usage. The present code is fully functional for nonlinear calculation of cross sections, 2D and 3D calculations on element level and 2D calculations for frames using the described 2 and 3 Node finite elements. For 2D cases, additionally, truss elements are implemented. Furthermore, the function of the potential energy is described using by a principle example with the discussion of the instabilities of the potential energy for non-linear systems. The structural optimization is addressed firstly on cross section level by applying the equilibrium conditions within the constrains, then the initial proposition of using a multi-objective optimization, for simultaneously minimizing the energy potential and an objective function was proved to be inadequate for structural optimization problems using the EIM, due to the inability to prove a stationary point on the unconstrained energy function. Instead, the nested method of the bilevel optimization was proposed and proved to be efficient.

Finally, the numerical code was tested and verified against existing ones, on cross section and element level. Selected optimization examples were presented, of which four on cross section level and four on element level. The cross section problems were from the sizing optimization and chosen in a way to correspond to the engineering practice. On element level, two of the examples were chosen as sizing problems and another two involved discrete shape optimization. The comparison with the analytical solution for the simply supported steel beam and for the arch with the theoretical catinery corresponded well. Additional accomplishment was the reduction of the weight of the truss bridge compared to previous studies which proved that this method could be competitive just as the FEM.

6.2 Conclusions

Structural optimization is generally good strategy for a favorable initial design of cross sections and systems in structural engineering. Often it leads in the right direction in doubtful situations when a decision for more than one parameter has to be made in the engineering practice. The two aspects of the structural optimization are the method used for securing equilibrium and the method for the optimization.

Although unconventional, the EIM method has proved to be an satisfying method for securing an equilibrium. It is highly efficient for nonlinear analysis of mechanical systems with low number of parameters due to the convexity of the potential energy and the high computation power offered by nowadays regular system configurations. For structural systems with relatively high number of variables, it may experience difficulties, obtaining the minimum of the potential; however commercial solvers and algorithms for convex programming has improved significantly in the recent history and therefore its potential in future application is promising. Currently the EIM method is only developed for line-like structures and cross sections, with only a peak into thin plates and shells, which is another limitation.

For the second aspect of the structural optimization, initially the multi-objective optimization was proposed. Using this optimization, only the region in the objective function space is defined, where the optimal solution is; however a stationary point of the potential can not be obtained; thus the equilibrium conditions are not met. Therefore the bilevel optimization is proposed, from which the most robust method is chosen, the nested one. This method is of high computational cost as for each iteration of the upper objective, a lower one has to converge; however for few design parameters the solution is obtained quickly. Additional issue that has to be addressed is the choice of the algorithm for the upper objective function, which is not always an easy task, since in most situations the properties of it are unknown; thus experience from engineering practice and good initial parameters is generally a must. The simplex algorithm has proved to be very efficient and relatively fast for few design parameters, typically for sizing problems, whilst the genetic algorithm for significant number of design parameters, where generally there is more then one minimum of the objective function, is superior; however since it is not deterministic it does not always converge to the stationary point. Therefore, as a reference, the hybrid algorithm, which obtain the first, usually satisfying, initial parameters using the genetic algorithm and then employing the simplex for finessing is recommended for discrete shape optimization. Compared to the equilibrium constrained optimization on cross section level, the nested one proved to be more robust and find a minimum with worse guess of the initial conditions.

Choosing another method can significantly decrease the computational effort for the bilevel optimization. There exists state of the art methods which apply the KKT conditions of the lower objective function as constrains in the upper one which results in a single objective functions, that are not reviewed within this work. Conclusively, it is hoped that structural optimization with EIM, will eventually prevail over the FEM method, especially for nonlinear analysis, since it does not affect the minimization of energy significantly, whilst in the FEM iterative procedure for every step has to be applied.

6.3 Scope for further work

The clear limitations of the nested optimization presented herein are the motivation for further work within this area. Choosing another solution methodology would significantly increase computational efficiency, which was not available in the chosen commercial package MATLAB. Numerous studies are currently focusing on the afore mentioned Mathematical Programs with Equilibrium Constrains (MPEC), which involves creating an artificial single objective function, in which constrains the stationary point of the lower objective function is included, and due to the convexity of total potential energy, this would be a perfect fit.

Further research could involve extension of the EIM method for plates, as there already have been previous studies for thin walled and shell structures and implementation of 3D frames. Additionally coupling this method with the standard FEM, for regions with high plasticity, such as concrete hinges, could utilize the advantages of both methods, the accuracy and efficiency of the FEM for linear analysis where iterative procedures does not have to be applied and the high precision of the EIM in plastic regions where only few parameters are unknown.

An extension to the presented code would involve: including 3D frames along with the truss elements, applying flexible bond on element and cross section level, predeformations by which prestressing and temperature effects could be modeled and Graphical User Interface (GUI) which would increase the usability.

Bibliography

[1] *Eurocode 2: Design of concrete structures.* International Federation for Structural Concrete, 2004.

[2] *Model Code 2010*, volume 2. European Committee for Standardisation, 2010.

[3] G.B. Allende. *Mathematical Programs with Equilibrium Constraints: Solution techniques from Parametric Optimization.* PhD dissertation, University of Twente, 2006.

[4] P. Anthony. Variational and approximate methods in applied mathematics. Lecture script (University of British Columbia), 2002.

[5] R.T. Marler; J.S. Arora. Survey of multi-objective optimization methods for engineering. *Structural Multidisciplinary Optimization*, 26:369–395, 2004.

[6] A.H. Arshian. Physically nonlinear fem analysis of reinforced concrete frames under extreme loading. Master's thesis, Bauhaus University, 2012.

[7] J.F. Bard. *Practical Bilevel Optimization - Alghorithms and Applications.* Kulwer Academic Publishers, 1998.

[8] P.A. Jensenl; J.F. Bard. *Operations Research Models and Methods.* John Wiley and Sons, 2003.

[9] K.J Bathe. *Finite Element Procedures.* Prentice-Hall, Inc, 1996.

[10] D.P Bertsekas. *Constrained Optimization and Lagrange Multiplier Method.* Athena Scientific, 1996.

[11] S.C. Chapra; R.P. Canale. *Numerical Methods for Engineers.* McGraw-Hil, 2010.

[12] A. Sinha; P. Malo; K. Deb. Evolutionary bilevel optimization. Presentation (Genetic and Evolutionary Conference 2013).

[13] M. Caramia; P. Dell'Omo. *Multi-objective Managment in Freight Logistics Increasing Capacity, Service with Optimization Alghorithams.* Springer, 2008.

[14] J. Dempe. Annotated bibliography on bilevel programming and mathematical programs with equilbrium constrains. *Optimization*, 52:333–359, 2003.

[15] S. Dong. Methods for constrained optimization. Project, 2006.

[16] E. Raue; H.-G Timmler; R. Garke. On the phyisically non-linear analyisis of cyclic loaded reinforced concrete cross-sections with mathematical optimisation. *Journal of Civil Engineering and Managment*, 15(2):189–195, 2009.

[17] A. Geletu. Solving optimization problems using the matlab optimization toolbox. Tutorial (TU-Ilmenau), 2007.

[18] C.T. Kelly. *Iterative Methods for Optimization*. Society for Industrial and Applied Mathematics Philadelphia, 2009.

[19] P.W. Christensen; A. Klarbring. *An Introduction to Structural Optimization*. Springer Science + Business Media B.V, 2009.

[20] C. Könke. Introduction into the finite element method. Lecture script (Bauhaus Univeristy - Weimar), 2002.

[21] T. Lahmer. Nonlinear optimization - an introduction. Lecture script (Bauhaus Univeristy - Weimar), 2014.

[22] J. Herskovits; A. Leontiev. An interior point technique for solving bilevel programming problems. *Optimization and Engineering*, 14:381–394, 2013.

[23] D.H. Wolpert; W.G. Macready. No free lunch theorems for search. Technical report, Santa Fe Institute, 1995.

[24] S. Marx. *Anwendung der matematischen Optimierung bei der geometrisch und physikalisch nichtlinear Analyse von Stahlbetontragwerken*. PhD dissertation, Bauhaus Univeristy - Weimar, 2000.

[25] J.A. Nelder; R. Mead. A simplex method for function minimization. *The Computer Journal*, 4(7):308–313, January 1965.

[26] P. Olney. Nonlinear analysis of composite cross-sections and elements using energy methods. Master's thesis, Bauhaus University, 2012.

[27] D.C. Kent; R. Park. Flexural members with confined concrete. *Structural Multidisciplinary Optimization*, 97:1969–1990, 1971.

[28] M.J.D. Powell. Algorithms for nonlinear constraints that use lagrangian functions. *Mathematical Programming*, 14:224–248, 1978.

[29] E. Raue. Nichtlineare querschnittsanalzse als optimierungsproblem. *Bautechnik 82*, pages 796–809, 2005.

[30] E. Raue. Non-linear analysis of composite cross-sections by non-linear optimisation. In *Proceedings 9th International Conference Modern Building Materials, Structures and Tehniques*, Vilnius, 2007.

[31] E. Raue. Non-linear analysis of composite cross-sections with pre-deformations. In *18th International Conference on the Application of Computer Science and Mathematics in Architecture and Civil Engineering*, Weimar, 2009.

[32] M.P. Saka. Shape optimization of trusses. *Journal of Structural Engineering ASCE*, 105:1155–74, 1980.

[33] E. Raue; H.-G Timmler; H. Schröter. Analytical modelling of retrofitted reinfoced concrete members with flexible bond. In *Concrete Repair, Rehabilitation and Retrofitting*, London, 2009. Taylor and Francis Group.

[34] E. Raue; H.-G Timmler; H. Schröter. Non-linear analysis of shells of revolution using mathematical optimisation. In *18th International Conference on the Application of Computer Science and Mathematics in Architecture and Civil Engineering*, Weimar, 2009.

[35] E. Raue; H.-G Timmler; H. Schröter. Non-linear analysis of the long-term behaviour of composite elements using mathematical optimization. In *Proceedings 9th International Conference Modern Building Materials, Structures and Tehniques*, Vilnius, 2010.

[36] H. Schröter. *Nichtlineare Analyse von Verbundelementen auf der Grundlage von Energieprinzipien unter Anwendung der mathematischen Optimierung*. PhD dissertation, Bauhaus Univeristy - Weimar, 2014.

[37] H.-P. Schwefel. *Numerical optimization of computer models*. Wiley, Chichester, 1981.

[38] K. Bryan; Y. Shibberu. Penalty functions and constrained optimization. Lecture script (Rose-Hulman Institute of Technology).

[39] A.E. Eiben; J.E. Smith. *Introduction to Evolutionary Computing*. Springer, 2007.

[40] S. Spreng. Identification of static equilibria via potential energy optimization. Master's thesis, Stuttgart University, 2010.

[41] M. Abramovitz; I.A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. United States Department of Commerce, 1965.

[42] C.L. Su. *Equilibrium Problems with Equilibrium Constrains: Stationarities, Alghorithams and Applications*. PhD dissertation, Stanford University, 2005.

[43] H.W. Kuhn; A.W. Tucker. Nonlinear programming. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492, Berkeley,California, 1951. University of California Press.

[44] S. Boyd; L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2009.

[45] A. Čyras; A. Borkowski; R. Karkauskas. *Theory and Methods of Optimization of Rigid-Plastic Systems Strategies*. Vilnius Technika, 2004.

[46] T. Weise. *Global Optimization Algorithms - Theory and Application*. 2009.

[47] R. Weitzmann. *Theory and Application of Optimization Strategies for the Design of Seismically Excited Structures*. Habilitation, Bauhaus Univeristy - Weimar, 2008.

[48] D.H. Wolpert and W.G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, April 1997.

[49] C.K. Soh; J. Yang. Optimal layout of bridge trusses by genetic algorithms. *Computer-Aided Civil and Infrastructure Engineering*, 13:247–254, 1998.

Appendix A

A.1 Finite elements implemented

The finite elements implemented are shown below. The shape functions were adopted from [36] and checked after [26] and [11]. The abbreviations are made after the number of nodes required for the Hermite polynomials. In case of 3D analysis the 2Node element has 11 and the 3 Node one, 17 degrees of freedom. For the 2D case the first one has 7 and the latter one 11. The 3Node and 2Node truss elements are without the transverse and rotational degrees of freedom i.e. only the Lagrange interpolation functions are used.

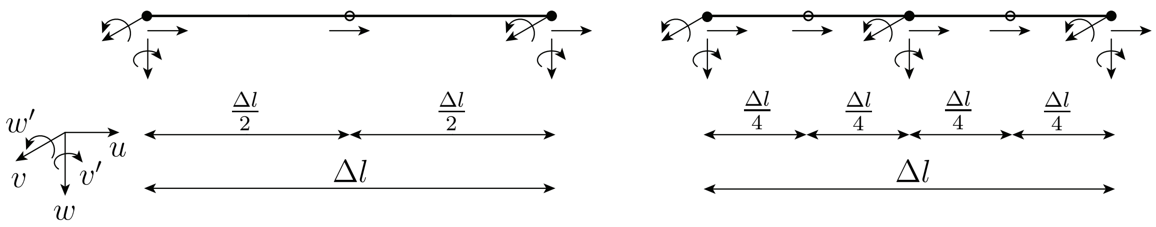


Figure A.1: Degrees of freedom for 2Node (left) and 3Node (right) element

2 Node element

Lagrange 2^{rd} order polynomial for longitudinal displacements:

$$u(x) = \left(1 - \frac{3x}{\Delta l} + \frac{2x^2}{\Delta l^2}\right) u_1 + \left(\frac{4x}{\Delta l} - \frac{4x^2}{\Delta l^2}\right) u_2 + \left(-\frac{x}{\Delta l} + \frac{2x^2}{\Delta l^2}\right) u_3, \tag{A.1}$$

$$u'(x) = \left(-\frac{3}{\Delta l} + \frac{4x}{\Delta l^2}\right) u_1 + \left(\frac{4}{\Delta l} - \frac{8x}{\Delta l^2}\right) u_2 + \left(-\frac{1}{\Delta l} + \frac{4x}{\Delta l^2}\right) u_3. \tag{A.2}$$

Hermite 3^{rd} order polynomial for transverse displacements in both directions (w(x) and v(x)):

$$w(x) = \left(1 - \frac{3x^2}{\Delta l^2} + \frac{2x^3}{\Delta l^3}\right) w_1 + \left(\frac{3x^2}{\Delta l^2} - \frac{2x^3}{\Delta l^3}\right) w_2 \tag{A.3}$$

$$+ \left(x - \frac{2x^2}{\Delta l} + \frac{x^3}{\Delta l^2}\right) w_1' + \left(-\frac{x^2}{\Delta l} + \frac{x^3}{\Delta l^2}\right) w_2', \tag{A.4}$$

$$w'(x) = \left(-\frac{6x}{\Delta l^2} + \frac{6x^2}{\Delta l^3}\right)(w_1 - w_2) + \left(1 - \frac{4x}{\Delta l} + \frac{3x^2}{\Delta l^2}\right) w_1' + \left(-\frac{2x}{\Delta l} + \frac{3x^2}{\Delta l^2}\right) w_2', \tag{A.5}$$

$$w''(x) = \left(-\frac{6}{\Delta l^2} + \frac{12}{\Delta l^3}\right)(w_1 - w_2) + \left(-\frac{4}{\Delta l} + \frac{6x}{\Delta l^2}\right) w_1' + \left(-\frac{2}{\Delta l} + \frac{6x}{\Delta l^2}\right) w_2'. \tag{A.6}$$

3 Node element

Lagrange 4^{rd} order polynomial for longitudinal displacements:

$$\begin{aligned} u(x) = &\left(1 - \frac{25x}{3\Delta l} + \frac{70x^2}{3\Delta l^2} - \frac{80x^3}{3\Delta l^3} + \frac{32x^4}{3\Delta l^4}\right) u_1 + \left(\frac{16x}{\Delta l} - \frac{208x^2}{3\Delta l^2} + \frac{96x^3}{\Delta l^3} - \frac{128x^4}{3\Delta l^4}\right) u_2 \\ &+ \left(-\frac{12x}{\Delta l} + \frac{76x^2}{\Delta l^2} - \frac{128x^3}{\Delta l^3} + \frac{64x^4}{\Delta l^4}\right) u_3 + \left(\frac{16x}{3\Delta l} - \frac{112x^2}{3\Delta l^2} + \frac{224x^2}{3\Delta l^3} - \frac{128x^4}{3\Delta l^4}\right) u_4 \\ &+ \left(-\frac{x}{\Delta l} + \frac{22x^2}{3\Delta l^2} - \frac{16x^3}{\Delta l^3} + \frac{32x^4}{3\Delta l^4}\right) u_5, \end{aligned} \tag{A.7}$$

$$\begin{aligned} u'(x) = &\left(-\frac{25}{3\Delta l} + \frac{140x}{3\Delta l^2} - \frac{80x^2}{\Delta l^3} + \frac{128x^4}{3\Delta l^4}\right) u_1 + \left(\frac{16}{\Delta l} - \frac{416x}{3\Delta l^2} + \frac{288x^2}{\Delta l^3} - \frac{256x^3}{3\Delta l^4}\right) u_2 \\ &+ \left(-\frac{12}{\Delta l} + \frac{152x}{\Delta l^2} - \frac{384x^2}{\Delta l^3} + \frac{512x^3}{\Delta l^4}\right) u_3 + \left(\frac{16}{3\Delta l} - \frac{224x}{3\Delta l^2} + \frac{224x^2}{\Delta l^3} - \frac{512x^3}{3\Delta l^4}\right) u_4 \\ &+ \left(-\frac{1}{\Delta l} + \frac{44x}{3\Delta l^2} - \frac{48x^2}{\Delta l^3} + \frac{128x^3}{3\Delta l^4}\right) u_5 \quad . \end{aligned} \tag{A.8}$$

Hermite 5^{rd} order polynomial for transverse displacements in both directions (w(x) and v(x)):

$$\begin{aligned} w(x) = &\left(1 - \frac{23x^2}{\Delta l^2} + \frac{66x^3}{\Delta l^3} - \frac{68x^4}{\Delta l^4} + \frac{24x^5}{\Delta l^5}\right) w_1 + \left(\frac{16x^2}{\Delta l^2} - \frac{32x^3}{\Delta l^3} + \frac{16x^4}{\Delta l^4}\right) w_2 \\ &+ \left(\frac{7x^2}{\Delta l^2} - \frac{34x^3}{\Delta l^3} + \frac{52x^4}{\Delta l^4} - \frac{24x^5}{\Delta l^5}\right) w_3 + \left(x - \frac{6x^2}{\Delta l} + \frac{13x^3}{\Delta l^2} - \frac{12x^4}{\Delta l^3} + \frac{4x^5}{\Delta l^4}\right) w_1' \\ &+ \left(-\frac{8x^2}{\Delta l} + \frac{32x^3}{\Delta l^2} - \frac{40x^4}{\Delta l^3} + \frac{16x^5}{\Delta l^4}\right) w_2' + \left(-\frac{x^2}{\Delta l} + \frac{5x^3}{\Delta l^2} - \frac{8x^4}{\Delta l^3} + \frac{4x^5}{\Delta l^4}\right) w_3', \end{aligned} \tag{A.9}$$

$$\begin{aligned}
w'(x) = &\left(-\frac{46x}{\Delta l^2} + \frac{198x^2}{\Delta l^3} - \frac{272x^3}{\Delta l^4} + \frac{120x^4}{\Delta l^5}\right) w_1 + \left(\frac{32x}{\Delta l^2} - \frac{96x^2}{\Delta l^3} + \frac{64x^3}{\Delta l^4}\right) w_2 \\
&+ \left(\frac{14x}{\Delta l^2} - \frac{102x^2}{\Delta l^3} + \frac{208x^3}{\Delta l^4} - \frac{120x^4}{\Delta l^5}\right) w_3 + \left(1 - \frac{12x}{\Delta l} + \frac{39x^2}{\Delta l^2} - \frac{48x^3}{\Delta l^3} + \frac{20x^4}{\Delta l^4}\right) w_1' \\
&+ \left(-\frac{16x}{\Delta l} + \frac{96x^2}{\Delta l^2} - \frac{160x^3}{\Delta l^3} + \frac{80x^5}{\Delta l^4}\right) w_2' + \left(-\frac{2x}{\Delta l} + \frac{15x^2}{\Delta l^2} - \frac{32x^3}{\Delta l^3} + \frac{20x^4}{\Delta l^4}\right) w_3',
\end{aligned} \tag{A.10}$$

$$\begin{aligned}
w''(x) = &\left(-\frac{46}{\Delta l^2} + \frac{396x}{\Delta l^3} - \frac{816x^2}{\Delta l^4} + \frac{480x^3}{\Delta l^5}\right) w_1 + \left(\frac{32}{\Delta l^2} - \frac{192x}{\Delta l^3} + \frac{192x^2}{\Delta l^4}\right) w_2 \\
&+ \left(\frac{14}{\Delta l^2} - \frac{204x}{\Delta l^3} + \frac{624x^2}{\Delta l^4} - \frac{480x^3}{\Delta l^5}\right) w_3 + \left(-\frac{12}{\Delta l} + \frac{78x}{\Delta l^2} - \frac{144x^2}{\Delta l^3} + \frac{80x^3}{\Delta l^4}\right) w_1' \\
&+ \left(-\frac{16}{\Delta l} + \frac{192x}{\Delta l^2} - \frac{480x^2}{\Delta l^3} + \frac{320x^3}{\Delta l^4}\right) w_2' + \left(-\frac{2}{\Delta l} + \frac{30x}{\Delta l^2} - \frac{96x^2}{\Delta l^3} + \frac{80x^3}{\Delta l^4}\right) w_3'.
\end{aligned} \tag{A.11}$$

A.2 Lobatto quadrature

The following relations are adopted from [36] and [41]. The integration over an interval [-1,1] is defined with the following relation:

$$\int_{-1}^{1} f(x)\,\mathrm{d}x = \frac{2}{n_p(n_p-1)}[f(1) + f(-1)] + \sum_{i=2}^{n_p-1} w_i f(x_i) + R_n, \tag{A.12}$$

where, n_p is the number of integration points, w_i are the weights and R_n is the remainder which is not known exactly and is 0 for integrating exact degree polynomial. The weights are obtained from tables. Derived for the element on Figure 4.10 with a $supp(f(x)) = \Delta l$:

Integration with 2 integration points for exact $deg(f(x)) = 3$

$$x_1 = 0,\ x_2 = \Delta l, \tag{A.13}$$

$$f(x) = (f(x_1) + f(x_2))\frac{\Delta l}{2}. \tag{A.14}$$

Integration with 3 integration points for exact $deg(f(x)) = 5$

$$x_1 = 0,\ x_2 = \frac{\Delta l}{2},\ x_3 = \Delta l, \tag{A.15}$$

$$f(x) = \left(\frac{1}{6}(f(x_1) + f(x_3)) + \frac{4}{6}f(x_2)\right)\Delta l. \tag{A.16}$$

Integration with 4 integration points for exact $deg(f(x)) = 7$

$$x_1 = 0,\ x_{2,3} = \left(1 \mp \frac{1}{\sqrt{5}}\right)\frac{\Delta l}{2},\ x_4 = \Delta l, \tag{A.17}$$

$$f(x) = \left(\frac{1}{6}(f(x_1) + f(x_3)) + \frac{4}{6}f(x_2)\right)\Delta l. \tag{A.18}$$

Integration with 5 integration points for exact $deg(f(x)) = 9$

$$x_1 = 0,\ x_{2,4}\left(1 \mp 1\sqrt{\frac{3}{7}}\right)\frac{\Delta l}{2},\ x_3 = \frac{\Delta;}{2};\ x_5 = \Delta l. \tag{A.19}$$

$$f(x) = \left(\frac{1}{20}(f(x_1) + f(x_5)) + \frac{49}{180}(f(x_2) + f(x_4)) + \frac{32}{90}f(x_3)\right)\Delta l. \tag{A.20}$$

Integration with 6 integration points for exact $deg(f(x)) = 11$

$$x_1 = 0,\ x_{2,5} = \left(1 \mp 1\sqrt{\frac{7}{21}(7 + 2\sqrt{7})}\right)\frac{\Delta L}{2},\ x_{3,4} = \left(1 \mp 1\sqrt{\frac{7}{21}(7 - 2\sqrt{7})}\right)\frac{\Delta L}{2}, \tag{A.21}$$

$$f(x) = \left(\frac{1}{30}(f(x_1) + f(x_6)) + \frac{14 - \sqrt{7}}{60}(f(x_2) + f(x_5)) + \frac{14 + \sqrt{7}}{60}(f(x_3) + f(x_4))\right)\Delta l. \tag{A.22}$$