

Structural Identification Problem

Aleksei Suvorov^{1, a}, Giulio Palama¹ and Ekaterina Lysiakova²

¹*Politecnico di Milano, Piazza Leonardo da Vinci 32, Milan, 20133, Italy*

²*Peter the Great St.Petersburg Polytechnic University, Polytechnicheskaya 29, St.Petersburg, 195251, Russia*

Abstract. The identification problem of the existing structures through the Quasi-Newton and its modification, Trust region algorithms is discussed. For the structural problems, which could be represented by means of the mathematical modelling of the finite element code discussed method is extremely useful. The nonlinear minimization problem of the L^2 norm for the structures with linear elastic behaviour is solved by using of the Optimization Toolbox of Matlab. The direct and inverse procedures for the composition of the desired function to minimize are illustrated for the spatial 3D truss structure as well as for the problem of plane finite elements. The truss identification problem is solved with 2 and 3 unknown parameters in order to compare the computational efforts and for the graphical purposes. The particular commands of the Matlab codes are present in this paper.

1 Introduction

The problem of definition of the current condition of the existing structure is always one of the most important problems, which arises during the life cycle of the building. An estimation of the material properties is an everyday work of the surveying firms and companies, which deal with the renovation and repair of the buildings.

There exist two approaches for the evaluation of the material properties. The first is a destructive way in which the consistency is interrupted. For instance, taking samples and consecutive testing in the lab or the well-known in-situ pulling test belong to the first category of the surveys. In the other hand the non-destructive techniques are also widely used in such services and moreover for the buildings of cultural heritage.

The method, which is described here can be applied in both cases as for the entire structure as well for the part of the interest. The field of application has a wide range: from the simple truss structures to the planar surfaces, and here we would describe the cases for the 3D truss and CST finite element code.

Depending on the scope of the survey it can be utilized for the evaluation of the condition of the damaged structures or the current load-carrying state.

Nomenclature

{}	vector
[]	matrix
3D	three-dimensional
CST	constant strain triangle

^a Corresponding author: catherinewoolf@mail.ru

2 The identification problem

In the general framework the identification problem can be extremely useful in the cases when the points of interest are actual stress state or the material properties.

The basic concept of the identification problem for the 3D truss structure consists in the procedure of application of the static load with the known value and the following estimation of the nodal displacements $\{u\}$. Due to the absence of the possibility to test the real structure, we simulate the measurement by the direct procedure of the Finite Element code. In this case we are able to obtain the displacements which are provoked by the certain load condition.

After the construction of the stiffness matrix and definition of the solution which is now and later called as known. We need to repeat the entire procedure again with inserting instead the known quantities the symbolic variables.

As the consequence, we can construct the vector of unknowns $\{u_{Unknown}\}$, which in each string contains the expression by means of symbolic variables, as it is written in the Equation (1).

$$\{u_{diff}\} = \begin{Bmatrix} f_1(E, \dots, E_n) \\ \cdot \\ \cdot \\ f_k(E, \dots, E_n) \end{Bmatrix} - \begin{Bmatrix} u_1 \\ \cdot \\ \cdot \\ u_k \end{Bmatrix} = \begin{Bmatrix} 0 \\ \cdot \\ \cdot \\ 0 \end{Bmatrix} \quad (1)$$

Finally, we can obtain the equality $\{u_{Unknown}\} - \{u_{Known}\} = \{0\}$, which describes the compatibility of the nodal displacements in the certain directions. In order to obtain the values of unknown parameters we need to apply the minimization procedure [10, 16].

3 Minimization problem

3.1 Least squares method

The basic formulation of the minimization problem can be defined through the least squares method, which finds the parameters that minimize the sum of the squares of the errors between experimental data $\{u_{Unknown}\}$ and model predictions $\{u_{Known}\}$. In the other words in our problem we are identifying material properties by means of formula (2):

$$MIN E = \frac{1}{2} \sum_{i=1}^k \left| \{u_{Unknown}\} - \{u_{Known}\} \right|^2 \quad (2)$$

Where, k is the number of load cycles,

E is an objective function to minimize.

It is should be noted that the previous expression is nonlinear and depends on the number of desired parameters to identify.

3.2 Advanced formulation of least squares method

In addition to the basic least squares method there exists more advanced formulation, which takes in account the variance-covariance matrix which takes in account the statistical information and denoting by x the desired parameters, so then the objective function $E(x)$ becomes:

$$E(x) = \frac{1}{2} \{u_{Unknown}(x) - u_{Known}\}^T \cdot [C(x)]^{-1} \cdot \{u_{Unknown}(x) - u_{Known}\} \quad (3)$$

The challenge of construction $[C(x)]$ matrix arises the problems of definition of statistical data based on the observations or calculated using a model, which for the sake of simplicity are omitted in the further calculations [14].

3.3 Unconstrained nonlinear optimization

If to denote by x our desired values then the problem can be formulated as $\text{MIN } E(x)$ which means that we are trying to identify a vector x that is a local minimum to a scalar function $E(x)$, without any restrictions to the place of x .

[A] In our case we can consider the unconstrained minimization problem to minimize $E(x)$ by means of trust region. Suppose you are at a point x in k -space and you want to improve, i.e., move to a point with a lower function value.

The basic idea is to approximate $E(x)$ with a simpler function q , which reasonably reflects the behaviour of function f in a trust region (neighbourhood) N around the point x . A trial step s is computed by minimizing (or approximately minimizing) over N . This is the trust-region sub-problem, $\text{MIN } \{q(s), s \in N\}$.

The current point is updated to be $x + s$ if $E(x + s) < E(x)$; otherwise, the current point remains unchanged and N , the region of trust, is shrunk and the trial step computation is repeated.

The sequence of steps in the unconstrained minimization using trust-region can be shown as:

- Formulate the trust-region sub-problem
- Solve $\text{MIN } \left\{ \frac{1}{2} s^T Hs + s^T g ; \text{such that } \|Ds\| \leq \Delta \right\}$

where g is the gradient of E at the current point x ,
 H is the Hessian matrix (the symmetric matrix of second derivatives),
 D is a diagonal scaling matrix,

Δ is a positive scalar, and $\| \cdot \|$ is the 2-norm.

This is done in order to determine the trial step s

- If $E(x + s) < E(x)$, then $x = x + s$
- Adjust Δ

These four steps are repeated until convergence. The trust-region dimension Δ is adjusted according to standard rules. In particular, it is decreased if the trial step is not accepted, which can be stated in the Equation (4).

$$E(x + s) \geq E(x) \tag{4}$$

The basic requirement of this optimisation procedure is the presence of the gradient of the function, which we wish to minimize [2].

[B] Another way to define the minimum of the function is the possibility to use Newton or Quasi-Newton algorithm. They require higher computation effort, but the derivative of function is not needed.

These methods are described by the formulation of Equation (5) and build up curvature information at each iteration to formulate a quadratic model problem of this form.

$$\text{MIN } \left\{ \frac{1}{2} x^T Hx + c^T x + b \right\} \tag{5}$$

where, H is the Hessian matrix (positive definite symmetric matrix),

c is a constant vector,

b is a constant,

The optimal solution for this problem occurs when the partial derivatives of the Equation (6) with respect to x go to zero

$$\nabla f(x^*) = Hx^* + c = 0 \tag{6}$$

And so the optimal solution x^* becomes of the form of the Equation (7) [15].

$$x^* = -H^{-1}c \tag{7}$$

3.4 Bayesian approach

This method does not provide a single value, for the identified parameter, but probability distribution of material properties is to be estimated. In Bayesian approach we are trying to identify the distribution of material properties X having an instance x . This vector is characterized by probability density function $\pi_x(x)$ [1].

The vector of measurements u_{Known} is assumed to be contaminated by uncertainty errors e^{meas} , stemming from the random variable Σ^{meas} . Then the true but unknown response is represented by the Equation (8).

$$u_{Unknown} = u_{Known} - e^{meas} \quad (8)$$

and we can denote by U_{true}^{meas} the random variable of true response, so that the Equation (8) can be translated by the Equation (9).

$$U_{true}^{meas} = u_{Known} - \Sigma^{meas} \quad (9)$$

Recalling the definition of x – our seeking random value, we can say state that the Equation (9) can be converted to the Equation (10), using the definition [7].

$$U_{true}^{model}(x) = u_{po}(x) - \Sigma^{model} \quad (10)$$

where, Σ^{model} is a random variable due to the approximations involved in the input parameters p at the modelling stage, $u_{po}(x)$ is the function of material properties and also other p input parameters [3]. Bayesian identification involves calculating, for given material properties $-x$, the likelihood that the true response is the same whether derived from the measurement or from the model.

$$E = U_{true}^{model} - U_{true}^{meas} = 0 \quad (11)$$

Such that the Equality (11) should be valid [8, 9].

4 Truss 3D finite element procedure

4.1 Code for the direct problem

In the problem solution we have used the code, which solves every kind of truss structures under the hypothesis of linear elastic and first order approach [13].

In order to simulate the measurements of the nodal displacements we need to run the direct procedure, which estimates the values by means of the equilibrium.

The models are drawn by means of a pre-processor software GMesh in which the user inserts the nodes and the rods of the truss structure. The drawn data are contained in a file with a .geo extension that can be read by Matlab like a text file.

Through the script read_input_analysis.m Matlab acquires from the file .geo the number of nodes, number of elements, Coordinates Vector and connectivity vector. Every node and every beam have an identification number.

Young Modulus and area of every beam, the number of restrained nodes and loaded nodes, the degree of restrained of every restrained node, the load on every loaded node are also required data.

Following the directions in Matlab Command Window, these data can be inserted manually and can be saved in a file “.mat” that can be loaded in order to run the analysis.

Once input analysis phase is concluded, the code starts to initialise vectors and matrixes that will be used in the computation.

The main part of the code is the assembly of stiffness matrix, obtained performing a loop over the elements.

It starts with the identification of the number of nodes and their coordinates then computes the length of the element and the transformation matrix.

The transformation matrix is a rotation matrix to transform a vector from local reference system to global reference system.

Let us consider a beam in the 3D description of the rotation matrix, represented by the Equation (12).

$$\begin{Bmatrix} U_1^L \\ U_2^L \end{Bmatrix} = \begin{bmatrix} S(\gamma)C(\theta) & S(\theta)S(\gamma) & C(\theta) & 0 & 0 & 0 \\ 0 & 0 & 0 & S(\gamma)C(\theta) & S(\theta)S(\gamma) & C(\theta) \end{bmatrix} \begin{Bmatrix} U_x^1 \\ U_y^1 \\ U_x^2 \\ U_y^2 \\ U_x^3 \\ U_y^3 \end{Bmatrix} \quad (12)$$

By C and S are denoted corresponding trigonometric functions *Sine* and *Cosine*.

Knowing the rotation matrix, we can write the stiffness matrix of the element in the global reference system. After that identifying the degrees of freedom of the element, the code assembles the global stiffness matrix K .

In order to consider the restrained nodes, the code computes the load vector and modifies the stiffness matrix after the assembling.

Defining an α parameter such as a big number and summing it in K matrix terms, it is possible to take into account the infinite rigidity given by restrains.

Finally, it is possible to obtain the solution of the linear system in terms of the vector $\{u\}$ representing the displacement field (Table 1).

Table 1. Code. Part 1.

```
%Solution of the linear system
u=K\F;
```

Knowing $\{u\}$ vector it is possible to calculate the reaction forces and the strain in every beam [4].

4.2 Code for the inverse identification problem

The identification procedure starts with the initialization of a vector of unknowns in which there are the Youngs modulus written in symbolic form (Table 2).

Table 2. Code. Part 2.

```
YoungUn=sym('E',[1 length(Young)]);%Initialization of vector of unknowns
```

After that, the same procedure of direct problem is performed until the solution of linear system, by doing this it is possible to obtain a vector of displacement that depends by the modulus of elasticity of the elements.

At this point, a vector of displacement as a function of Young modulus and the numerical results given by the direct procedure are available.

In order to find the value of Young modulus it is possible to state the Equation (13) should be admitted.

$$\{uUnknown\} - \{uKnown\} = \{0\} \quad (13)$$

that in explicit form can be converted to Equation (14).

$$\{u_{diff}\} = \begin{Bmatrix} f_1(E, \dots, E_n) \\ \cdot \\ \cdot \\ f_k(E, \dots, E_n) \end{Bmatrix} - \begin{Bmatrix} u_1 \\ \cdot \\ \cdot \\ u_k \end{Bmatrix} = \begin{Bmatrix} 0 \\ \cdot \\ \cdot \\ 0 \end{Bmatrix} \quad (14)$$

By estimating the norm in L^2 of the vector $\{u_{diff}\}$ it is possible to obtain the function to minimize [4, 17, 18].

5 CST finite element procedure

5.1 Code for the direct analysis

The code solves linear elastic analyses for plane problems in plane strain or plane stress formulations. The models are drawn by means of a pre-processor software GMesh in which the user inserts the nodes and the contour of the section. The drawn data are contained in a file with a .geo extension that can be read by Matlab like a text file. The finite element mesh is obtained by using GMesh and is stored in a .msh file [20].

From a text file prepared by the user, and through the script read_input_analysis.m, Matlab acquires information about the material, such as Young modulus and Poisson coefficient, and obtains boundary conditions [21].

Once vectors and matrix are initialized, the main goal of the code is the assembly of stiffness matrix [19], obtained performing a loop over the elements.

Then, the element stiffness matrix is computed by means of a script function stiff_t3.m which takes in account the choice of plane strain or plane stress [4].

The last step before the solving system is to consider the boundary conditions and the nodal forces.

Finally, it is possible to obtain the solution of the linear system in terms of the vector u representing the displacement field. Stresses are computed as post-processing [4].

5.2 Code for the inverse identification problem

The identification procedure consists in a repetition of the direct procedure with an initialization of an unknown elastic modulus written in symbolic form and inserted into the element stiffness matrix script function, which becomes stiff_t3un.m.

After that, the same procedure of direct problem is performed until the solution of linear system, by doing this it is possible to obtain a vector of displacement that depends by the modulus of elasticity of the elements (Table 3).

Table 3. Code. Part 3.

```
%% solving linear system
disp 'solving linear system';
uUn=KKun\Ff;
```

At this point, a vector of displacement as a function of Young's modulus and the numerical results given by the direct procedure are available.

In order to find the value of Young modulus it is possible to state the Equation (13) should be admitted, that in explicit form leads to the Equation (14).

By doing the norm in L^2 of the vector $\{u_{diff}\}$ it is possible to obtain the function to minimize (Table 4) [16,11,12].

Table 4. Code. Part 4.

```
LML=uUn-u;
NorLML=norm(LML);
```

5.3 Minimization procedure using Matlab Toolbox

In order to estimate the desired values the unconstrained minimization procedure was used. The chosen algorithm for the solution is Quasi-Newton (Table 5) [6].

Table 5. Code. Part 5.

```
options = optimoptions(@fminunc,'Algorithm','quasi-newton');
x0 = [150000,150000,150000]; % Starting initial guess
options.TolFun = 1e-100;
options.TolX = 1e-100;
%fun2min_...=function to minimize
[ELA,fval,exitflag,output] = fminunc(@fun2min_trepiediZ,x0,options)
```

From the previous steps by the solution of direct and inverse problems we were able to obtain the function which we want to minimize. It is required to insert the function into “fun2min_...m” and run the minprocQN file [5, 6, 12].

6 Numerical results

6.1 Truss 3D

Considering a structure composed by three beams and four nodes at the Figure 1 (a), it is possible to illustrate the way of computations.

The coordinates of nodes are

$P_1=(333,333,333)$ $P_2=(0,0,0)$ $P_3=(1000,0,0)$ $P_4=(0,1000,0)$.

The value of nodal load applied to the top is $V=10000000$ N

The elastic parameters are $E1=2.0e+05$ $E2=3.0e+05$ $E3=2.5e+05$

All quantities are expressed in the SI units (N, mm).

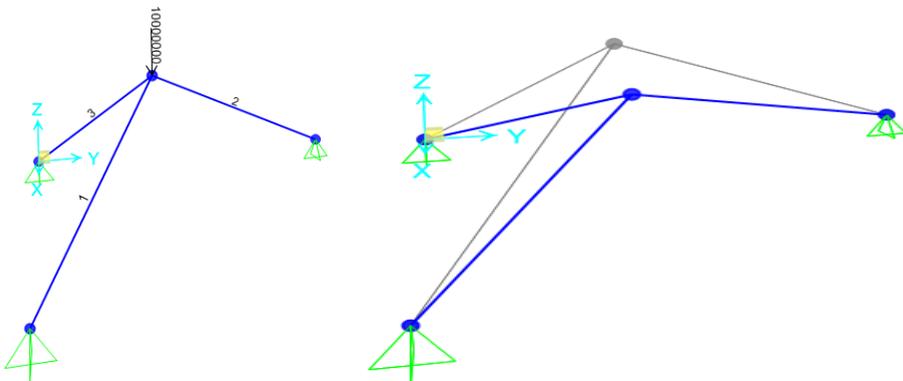


Figure 1. a) 3D Truss model in SAP; b) Obtained displacement in SAP.

Performing a structural linear elastic analysis of this model with the commercial finite elements software SAP2000, the following result in terms of displacements are obtained (Figure 2).

	Joint Text	OutputCase	CaseType Text	U1 mm	U2 mm	U3 mm	R1 Radians	R2 Radians	R3 Radians
▶	1	DEAD	LinStatic	2,175075	1,157613	-8,231133	0	0	0
	2	DEAD	LinStatic	0	0	0	0	0	0
	3	DEAD	LinStatic	0	0	0	0	0	0
	4	DEAD	LinStatic	0	0	0	0	0	0

Figure 2. Results of calculation in SAP.

At this point the displacement field is related the following deformed shape is displayed at Figure 1 (a). The results obtained through the Matlab direct code are collected to the Table 6 and the deformed shape is displayed at Figure 3.

Table 6. Results of Matlab direct procedure.

Node	U1 [mm]	U2 [mm]	U3 [mm]
1	2.4852	1.3301	-6.7586
2	0	0	0
3	0	0	0
4	0	0	0

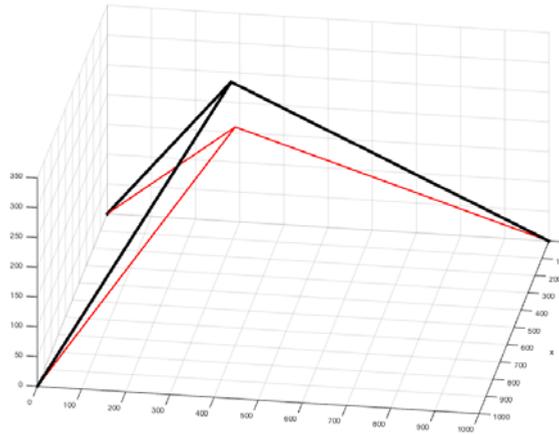


Figure 3. Deformed configuration in Matlab;

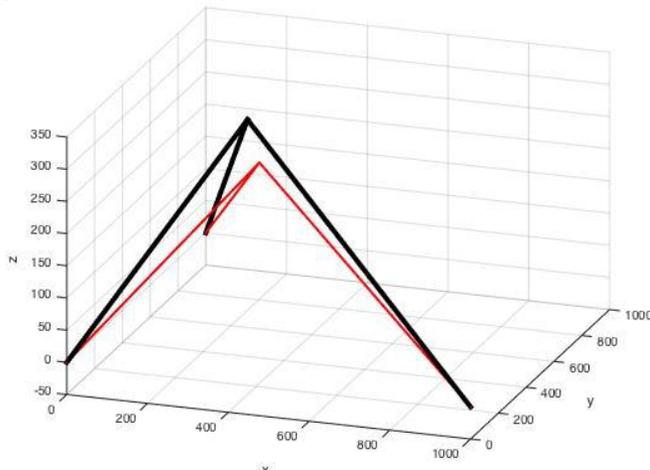


Figure 4. Deformed configuration in Matlab.

The vector $\{Uknown\} = \{2.4852, 1.3301, -6.7586\}$ mm is obtained.

Then, using the inverse procedure it is possible to get a function, which depends on three elastic modules $E1 E2 E3$. This is exactly the function, which is needed for the further computations and estimation of the unknowns in the next step.

In order to use the Quasi-Newton algorithm of the Matlab Optimization Toolbox it is needed to set the initial guess. For the physical problems the range of the output values can be predicted, that is why starting guess $[1.5e+05 \ 1.5e+05 \ 1.5e+05]$ was chosen. The results of the solution of our nonlinear minimization problem by means of Matlab code are listed at the vector $ELA = [2e+05 \ 3e+05 \ 2e+05]$ are obtained.

Finally, it can be easily noticed that vector ELA of the desired parameters perfectly matches to the vector of the inserted elastic moduli during the direct simulation procedure.

6.2 Truss 3d with 2 unknown parameters

Using the geometric parameters of the problem discussed before we can solve the problem with two unknowns.

Setting the initial vector of elastic moduli for the direct procedure as $[2, 0 \cdot 10^5, 3, 0 \cdot 10^5, 2, 5 \cdot 10^5]$ we obtain the deformed configuration shown at Figure 4.

The vector of simulated measurements is presented at the Table 7.

Table 7. Results of Matlab direct procedure.

Node	U1 [mm]	U2 [mm]	U3 [mm]
1	2.4852	1.3301	-6.7586
2	0	0	0
3	0	0	0
4	0	0	0

For the inverse procedure we set the symbolic vector of unknowns as $[E1, E2, 2, 5 \cdot 10^5]$ and solve the problem with two unknowns.

Obtaining the function which we need to minimize it is possible to plot the 2D plot depicting the point of the minimum and 3D surface (Figure 5) from which is clearly seen the coordinates of extreme point $E1=2, 0 \cdot 10^5$ and $E2=3, 0 \cdot 10^5$.

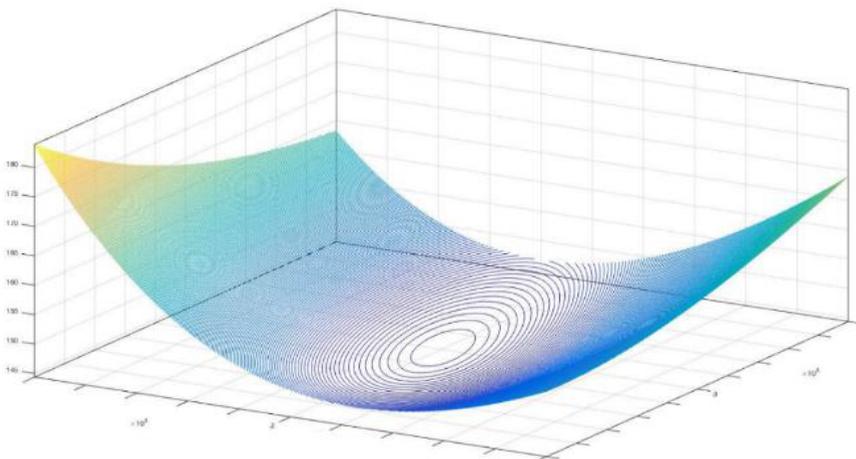


Figure 5. 3D surface in E1 and E2 axis.

By means of the inverse procedure of Matlab code we can observe in the Figure 9 the value of the function that goes to the minimum point and the number of iteration.

Matlab performs 32 iteration to obtain the values of the modules of elasticity which are slightly different from the exact ones $E1=2.0005 \cdot 10^5$ and $E2=3.0001 \cdot 10^5$.

The variation between exact value of the input and the result of minimization can be provoked by the threshold for the iteration procedure.

6.3 CST Finite element

Given the high requirement of computational resources needed for inverting the system in symbolic manner, the CST finite element identification problem has been performed on simple structure and with a very reduced number of finite elements. Nevertheless, after several tests performed on different samples, a rather good result has been observed.

Let us consider a concrete block with unknown elastic modulus subjected to a vertical distributed load $q=100 \text{ N/mm}$, shown at the Figure 6.

In the real case procedure, it should be supposed to measure directly the nodal displacements under the load condition. Although, assuming a Poisson ratio 0.3 and a square cross section $400\text{mm} \times 400\text{mm}$, it is possible to obtain the displacements field from the direct CST procedure by imposing a value to the Young modulus (in this example $\text{YOUNG}=24500 \text{ N/mm}^2$).

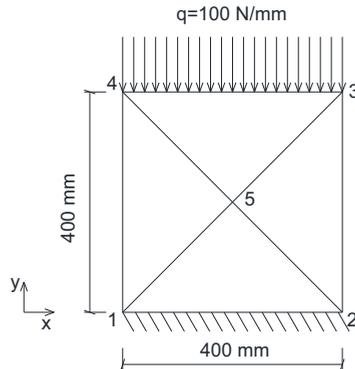


Figure 6. CST example.

Therefore, by means of a plane strain analysis the following displacement field is obtained in Table 9.

Table 8. Displacement field of CST model.

Direction	Value [mm]
U1x	0.0000
U1y	0.0000
U2x	0.0000
U2y	0.0000
U3x	0.4122
U3y	-1.3895
U4x	-0.4122
U4y	-1.3895
U5x	0.0000
U5y	-0.5802

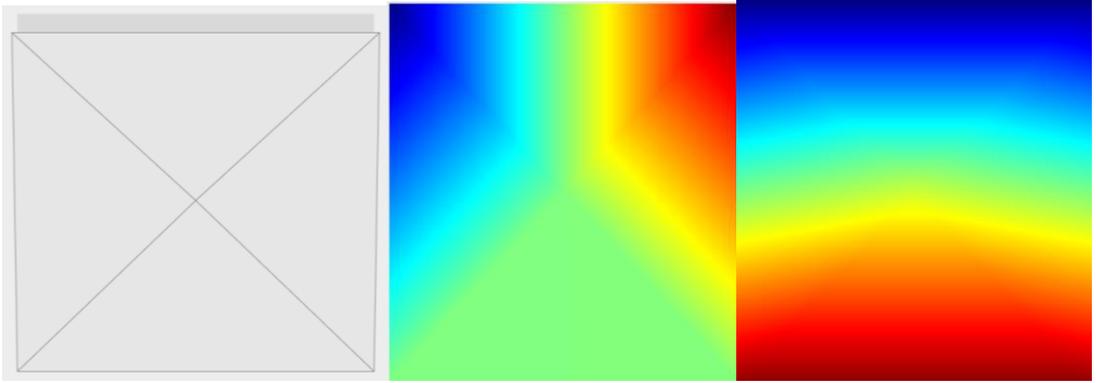


Figure 7. a) Deformed configuration CST. b) Displacement diagram in the direction u_1 and u_2

At this point, by means of the inverse code it is possible to obtain a displacement vector as function of the unknown parameter E .

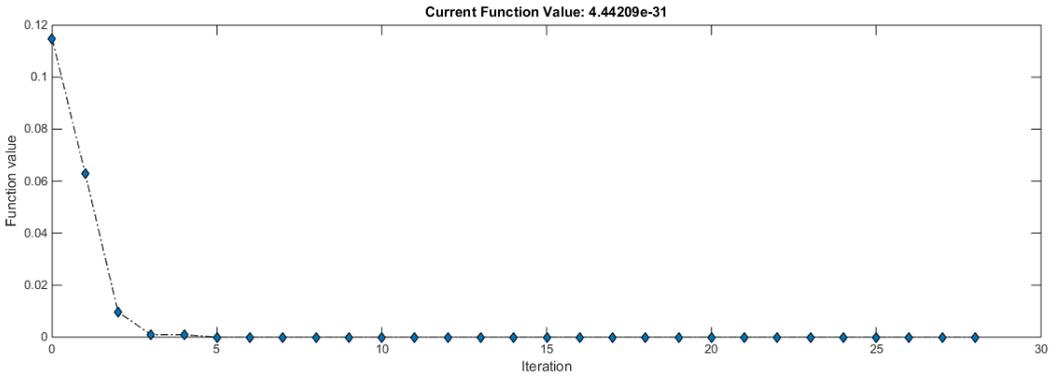


Figure 8. Iterations in CST model.

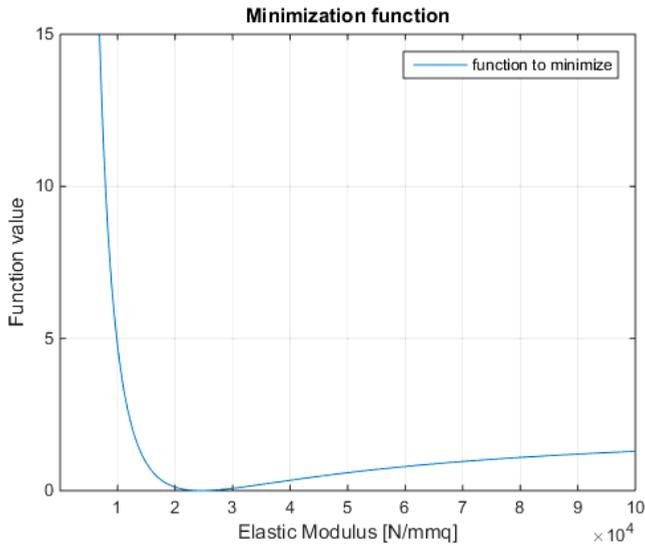


Figure 9. The function to minimize.

Using the Quasi-Newton algorithm of the Matlab Optimization Toolbox with a starting guess = [20000] and after 28 iterations (Figure 8), the correct value of the desired parameter ELA = [2.4500e+04] is obtained through the minimisation of the function shown at Figure 8.

7 Conclusions

In the real cases of the truss structures, the parameters of the most components are defined from the very beginning. The solution is the subject for identification is the matter several parameters. For instance having a complex structure as the object for survey we can definitely name the values related to the substructures. At least the groups of elements, which have the same properties can be defined. What is important is to mention that the structures that have uniform or classified distribution of sectional and material parameters require lower computational effort. This can significantly decrease the number of unknowns and as the consequence the complexity of the problem.

During the testing procedures of the Matlab code, we tried to run the structure with great number of rods, where each parameter is different and unknown and we mentioned that it requires the high computational capacity in order to construct the matrix of unknowns.

The surveying procedures can vary depending on the possibility to achieve the important parts of the structure and the used equipment. Sometimes the measurements are easier to perform in the terms of strains, in this case, the slight modification in the procedure are required, but the philosophy of the solution remains the same.

References

1. C. Gogu, R. Hatikat, Introduction to the Bayesian Approach Applied to Elastic Constants Identification. *AIAA JOURNAL* **48** 5 (2010)
2. J.J. Moré, and D.C. Sorensen, Computing a Trust Region Step. *SIAM Journal on Scientific and Statistical Computing* **3** 553–572 (1983)
3. J. Idier, *Bayesian Approach to Inverse Problems* 392 (Wiley-ISTE, Paris, 2008)
4. Alberto Corigliano, Alberto Taliercio, *Meccanica Computazionale: Soluzione del Problema Elastico Lineare* (Esculapio Editore, Italy, 2006)
5. The MathWorks. *Optimization Toolbox: User Guide* 634 (Natick, 2015)
6. J. E. Dennis, J. M. Jorge, Quasi Newton Methods, Motivation and Theory. *SIAM Review* **19** 46-89 (1977)
7. T. Marwala, S. Sibisi, Finite Element Model Updating Using Bayesian Framework and Modal Properties. *Journal of Aircraft* **42** 275-278 (2005)
8. K. F Alvin, Finite element model update via Bayesian Estimation and minimization of dynamic residuals. *AIAA Journal* **35** 879-886 (1997)
9. C Gogu, Facilitating Bayesian Identification of Elastic Constants through Dimensionality reduction and response surface methodology. Ph.D. Dissertation. University of Florida. *Gainesville, FL, and Ecole des Mines de Saint Etienne* (France, 2009)
10. M. S. Agbabian, S. E Masri, R. K. Miller, T. K. Caughey, System identification approach to detection of structural changes. *J. Engrg. Mech., ASCE* **117**(2) 370-390 (1991)
11. M. R. Banan, K. D. Hjelmstad, Parameter estimation of structure from static response I. Computational aspects. *J. Struct. Engrg., ASCE* **120**(11) 3242-3258 (1994)
12. M. R. Banan, K. D. Hjelmstad, "Parameter estimation of structures from static response. II. Numerical simulation studies. *J. Struct. Engrg., ASCE* **120**(11) 3259-3283 (1994)
13. M. Baruch, I. Y. Bar-Itzhack, Optimal weighted orthogonalization of measured modes. *AIM J* **16**(4) 346-351 (1978)
14. A. Bjorck, Numerical Methods for Least Squares Problems. *Society for Industrial and Applied Mathematics* (Philadelphia, 1996)
15. C.L. Lawson, R. J Hanson, *Solving Least Squares Problems* (NJPrentice Hall, Englewood Cliffs, 1974)

16. A. Tarantola, Inverse. Problem Theory and Model Parameter Estimation. *Society for Industrial and Applied Mathematics* (Philadelphia, 2005)
17. J. Isenberg, Progressing from Least-Squares to Bayesian Estimation. *Proceedings of the ASME Design Engineering Technical Conferences* 71–82 (American Society of Mechanical Engineers. New York, 1979)
18. O.C. Zienkiewicz, R.L. Taylor, *The Finite Element Method for Solid and Structural Mechanics* (Butterworth-Heinemann, 2013)
19. A. Munjiza, E. Rougier, E. E. Knight, *Large Strain Finite Element Method: A Practical Course* (Wiley, 2015)
20. S. S. Rao, *The Finite Element Method in Engineering* (Butterworth-Heinemann, 2010)
21. A.J.M. Ferreira, *MATLAB Codes for Finite Element analysis. Solids and Structures* (Portugal, 2009)