

## The Method of Transferring Topology Optimization Results Directly to the CAD System Database

Konrad Łyduch<sup>1\*</sup>, Jacek Łukaszyk<sup>1</sup>, Michał Nowak<sup>1</sup>

<sup>1</sup> Faculty of Mechanical Engineering, Poznań University of Technology, ul. Jana Pawła II 24, 61-139 Poznań, Poland

\* Corresponding author's email: [konrad.lyduch@put.poznan.pl](mailto:konrad.lyduch@put.poznan.pl)

### ABSTRACT

The article presents a method of transferring topological optimization results directly to CAD system. The developed method is based on sequential removal and addition of material, ignoring lesser fragments of the model. The input data includes coordinates of nodes imported from the model of finite element method optimized object. The algorithm is able to recognize geometry patterns during material addition and subtraction. Based on the results of the recognition process, individual 3D features are placed in CAD design tree (like „Feature Manager” design tree in Solidworks, used in the procedure described in the article below). Thus, the described algorithm allows for convenient and quick editing of transferred geometry of the optimized object directly in the CAD environment. The algorithm was tested on various examples of 2D and 3D models. The execution code of the presented method was written in Python programming language, and the macro for the CAD program was written in VB.NET. The proposed solution is independent of the optimizer used.

**Keywords:** topology optimization, CAD, transferring results, three-dimensional structures, automated system.

### INTRODUCTION

Technological progress has enabled implementation of issues previously formulated only from a theoretical standpoint. One such issue in mechanical engineering is topological optimization. Widespread dissemination of this technology, however, has resulted in emergence of new problems related to it.

A key issue is how to reproduce topology optimization result in CAD environment. Optimizers currently available make it possible to view topological optimization results and export them to STL format. It is a universal file, i.e. one that can be reproduced in any CAD program or even in applications available directly in the Windows operating system. However, such models do not have a feature tree to shape and edit the appearance of the solid. The problem is the STL model, which describes surfaces, while a solid type model is

best suited to editing the model. Also, presence of an operation tree is highly desirable.

On the basis of literature search, a triple approach can be distinguished towards the issue of transferring topological optimization result to the CAD system, performed after the optimization process (post-processing). First is a simple extrusion of planar (2D) elements, second is surface operations and variable profile extrusions, and third is the use of beams and bars for spatial (3D) elements, as in skeletonization method. At the same time, there are also methods that focus on obtaining a result directly in topological optimization process that could be easily reproduced in a CAD program.

Shape making in two-dimensional space has been described by Lin and Chao [1]. In this paper the data are parameterized from a 2D photo, then the external contour of the model is searched, empty spaces inside the object are detected which are finally adjusted to predefined shape templates. The templates were determined by measuring a

distance of contour nodes from gravity centre for several characteristic geometries. In total, seven templates were created for: circle, rectangle, triangle as well as several triangle-rectangle combinations. In this way, the complexity of material cutting operation was reduced to a few simple geometric figures [1].

The above-mentioned idea was further developed in the paper by Lin and Chou [2]. The number of predefined shape patterns was limited to only four, but at the same time corrections were made to the algorithm, thanks to which the detected holes are automatically expanded to maximum dimensions. The enlarged holes do not intersect nor do they intersect the outer contour of the geometry. Where possible, straight lines are used instead of B-spline curves, which would impede subsequent interference analysis. The presented solution combines topological optimization with shape optimization [2].

An attempt was made to integrate topological optimization with CAD environment by using a universal IGES format. It is readable in any available CAD/CAM system. This is yet another method that is performed after the topological optimization (post-processing). The algorithm is based on two-dimensional topological optimization solutions presented using a black and white colour palette. In the input JPG file, the edges of black sketches, i.e. solid material, are detected, then B-spline curve is calculated, and finally, based on specified curves, an IGES file is created. The algorithm has been equipped with a graphic user interface (GUI), which makes it possible, among others, to define the degree of smoothness of the created splines. The described method can be used in milling machine manufacturing or in production of piezoelectric microtransducers [3].

The concept of using surface operations along with lofting is described in the article by Tang and Chang [4]. The authors first converted the geometry obtained as a result of topological optimization into smooth and parameterized B-spline curves and surfaces. The surface smoothing effect was obtained by averaging boundary nodes/model outline and an improved least squares method. A final model in the CAD environment was created by extruding the material along B-spline surface, and then cutting voids in the resulting geometry using Boolean operations. The Solidworks program was used as CAD environment with API tools provided by the manufacturer [4].

Scientists also referred to this method in their deliberations on three-dimensional interpretation of topological optimization results. That is, extracting key cross sections and sweeping material along them. Higher accuracy of mapping the model requires an adequately larger number of selected cross-sections [5].

Another approach to three-dimensional reconstruction of topological optimization results was used in the work by Larsen and Jensen [6]. Here, two approaches to the problem were combined: two-dimensional [1] and three-dimensional [4]. Similarly, also the determination of significant solid cross sections was used. However, when recognizing cross-section shape, the following shapes were used: a circle, a triangle and a tetrahedron. Recognition was carried out by measuring the distance from any reference point in relation to the averaged points of the polar map, and thus the results were assigned to appropriate figures. Final shapes were achieved by cutting the material from solids, i.e. in the state prior to topological optimization was applied [6].

The third concept, skeletonization, is presented in several articles [8, 9]. In one of them, written by Cuillière et al. [7], triangle mesh resulting from optimization is smoothed and on its base a network of “skeleton” curves is created. The final result is a truss consisting of beams. The structure obtained with this method was less stiff in relation to the model exported directly from the optimizer.

There are software packages that facilitate a process of recreating topological optimization results from scratch. These include SolidThinking, which is an optimizer equipped with PolyNURBS tool that uses mechanics similar to skeletonization process. In one paper, the authors used a program designed to recreate the Geomagic Design X scan to obtain a CAD model of an optimized mechanical bracket [10].

Based on the above articles, it can be observed that the topic of automating interpretation of topological optimization results is still of great interest. However, each of the presented solutions has some limitations: either it works only for 2D planes, or uses only a beam structure, or the obtained model is not convenient to modify. Moreover, skeletonization can mainly be applied to optimization results with a fixed low density factor, which forces the optimized model to be more truss-like [7]. It is true that there is software available to facilitate the reconstruction of optimized models, but it merely

facilitates the process that must be carried out from start to finish by the user. Using the experience of the presented authors, a solution was prepared whose main purpose was to automatically transfer optimization results to one of popular CAD software packages as a solid model with an operation tree.

### MOTIVATION AND PURPOSE OF ALGORITHM CREATION

The presented algorithm is designed to transfer the results of topological optimization to CAD system. The goal is to solve several problems. First, at the very beginning, the transferred model is to consist of an operation tree that will enable quick editing of already existing operations. Second, it is supposed to be an automated solution. Third, the model must allow further modifications in order to accommodate fresh cutting or adding material.

For the above assumptions to be achieved, it was assumed that the obtained model will be of the “solid” type. Since the operation tree is to be present, so-called universal formats, such as IGES or STEP had to be omitted. To transfer the optimized model to CAD environment, Solidworks as the target program and API tools were used to create an appropriate macro. It is possible to write macros for other CAD programs, but due to time-consuming nature of such a process, only one program was selected.

The aim of the discussed method is to provide the user with a model that he will be able

to modify according to his needs. Regardless of technological process used according to which the model will be produced, whether milling, casting or 3D printing, the user will be able to independently, conveniently and efficiently process a transferred topological optimization result. The presented method aims to partially relieve the user of tedious process of manual reconstruction of the entire topological optimization result. Thanks to this, it will be possible to start from the result that appeared directly in the optimizer. This will allow the engineer to save time, which otherwise would have to be spent on recreating the entire model.

Two programming languages, Python and VB.NET, were used to create the code of the presented algorithm. Python’s popularity has grown rapidly in the past 8 years. According to Tiobe index in October 2021, Python was used by 11.27% users, which placed it in the first position in terms of the most popular languages [11]. Also according to Stack Overflow users it was the most popular programming language [12]. Python is characterised by ease of use and by a large number of libraries that can be used by scientists from various fields.

Prior to Python’s rise in popularity, Visual Basic - VB and VB.NET was considered the simplest programming language. Currently, it is not as popular as 17 years ago, when it was used by as many as 8.6% users, and now only 0.75%, according to PYPL website [13]. It is one of the few languages that can be used to write a Solidworks macro, right next to C#, C++ / CLI.

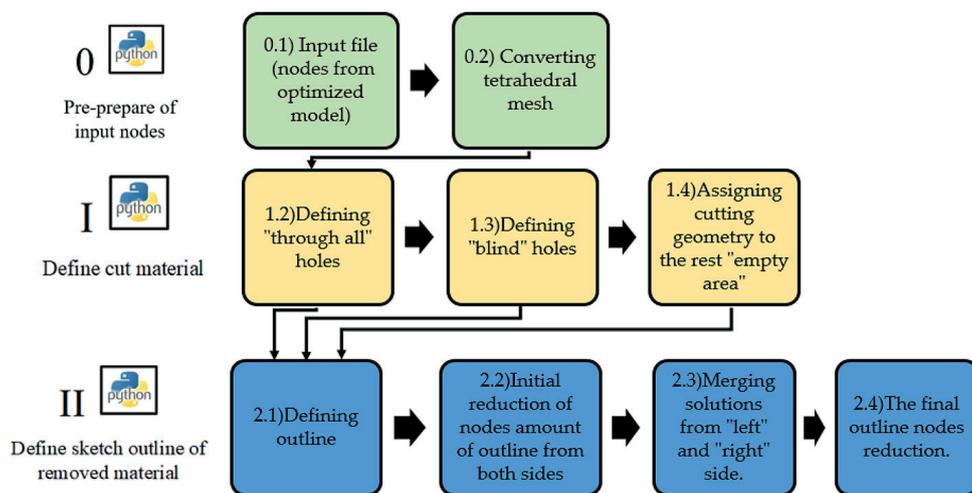


Fig. 1. Graphic diagram of the part of the algorithm responsible for recognizing the geometry to cut (written in Python)

## TOPOLOGICAL OPTIMIZATION RESULTS AS INPUT DATA

Input data includes coordinates of nodes imported from a Finite Element Method model of the optimized object. The information must be saved in a TXT text file and is available from optimizers such as Altair Hyperworks and Siemens Femap. There is no need to process the results in the optimizer before the data export. In the case of CAD, Solidworks or Inventor applications, the model of the finite element method of the optimized object must be defined with external software, e.g. Gmsh. Moreover, information about a defined mesh size, in an optimized model, is required to run the algorithm.

## GEOMETRY RECOGNITION ALGORITHM

The algorithm can be divided into two parts: reconnaissance and execution. The first one is responsible for processing the nodes so as to separate only those needed for making outlines of sketches intended for cutting operations. This part is implemented in the Python programming language (Figure 1). The second part, so-called execution, is a macro written to recreate the model in Solidworks. The processed data from the previous part of the algorithm is read and a solid model is made in the CAD environment. The macro has been written in VB.NET (Figure 4).

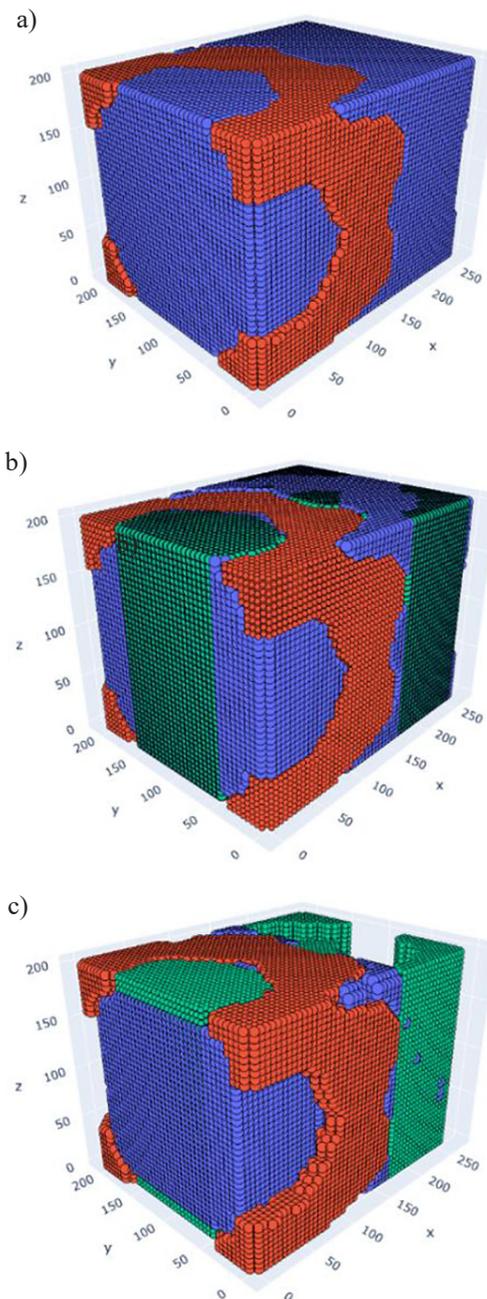
### Step one - identify the material that will be removed

First, nodes that represent through-all holes in XYZ axes are determined. Then the selected nodes are removed so that they are not included in further processing. Before proceeding to the next stage, some of the removed nodes are put back. At this stage, the goal is to absolutely remove only those nodes that actually need to be cut along the entire axis of the model. This means nodes that, when cut, do not leave any significant adjacent sets of nodes on a two-dimensional (2D) plane perpendicular to the cut axis. For example, for the Z axis, this means the XY plane (Figure 2).

In the second stage, the model fragments that will be removed by blind holes are identified in XYZ axes. After defining removable nodes, they are initially removed from the pool and then completed similarly as in the previous step.

In the last, third step, all other nodes that are in larger clusters are arranged in groups. Then each cluster is identified as a prism or pyramid. Target cut geometry is selected so as to best fit the shape of a given cluster of removable nodes.

In the final phase of the first and second stages, the nodes are replenished. As a result, it



**Fig. 2.** The main idea behind recognition algorithm - sequential subtraction and addition of material, omitting insignificant fragments of the model. (a) Condition prior to applying delete operation on the model (red); (b) material removed by a Z-axis through-all cut (green). (c) Filling the space with removable material (blue)

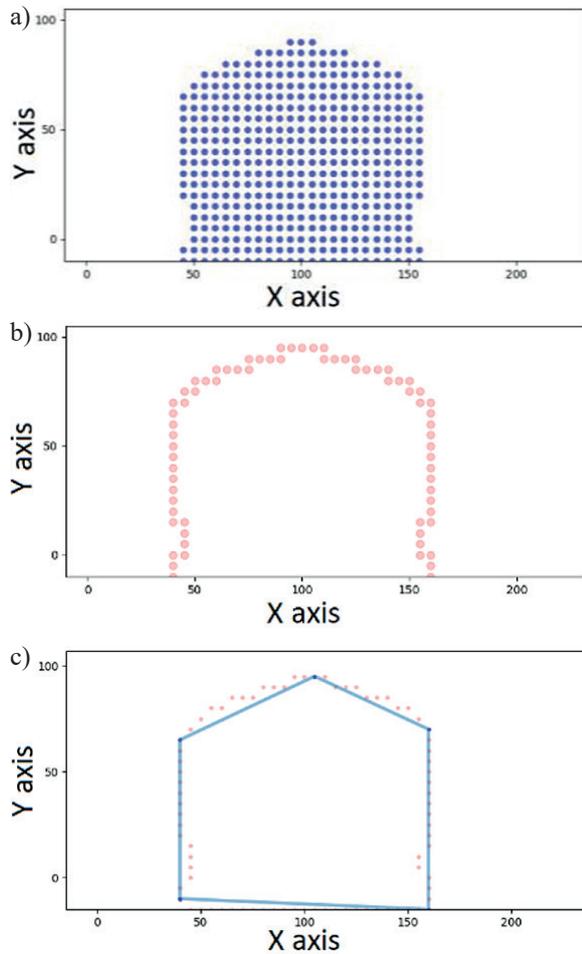
is easier in the second and third steps to correctly identify a smaller number of larger geometries to be cut, rather than performing multiple minor material removal operations.

Some studies, focused on the recognition of shapes i.e. shafts [14, 15]. However, methods used there assumed that processed models were

defined with normalized dimensions. In the case of topology optimization, results include many, different and irregular shapes, which is why these methodology would not work in the presented transferring algorithm [14, 15].

### Step two - define nodes and sketches of external opening outlines

In the last step, the predefined removable nodes were reduced to the number that is necessary for the correct representation of the outline shape of the sketches (Figure 3). A two-dimensional (2D) sketch consists of line-type curves.

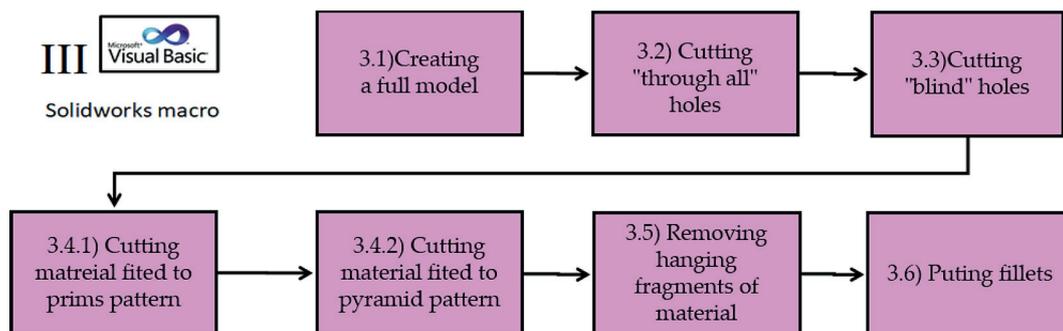


**Fig. 3.** (Step 2.1) Model simplification (operation on 2D cross-sections, based on the results of topological optimization): (a) input, removable nodes, (b) contour recognition, and (c) shape simplification and creation of a 2D sketch

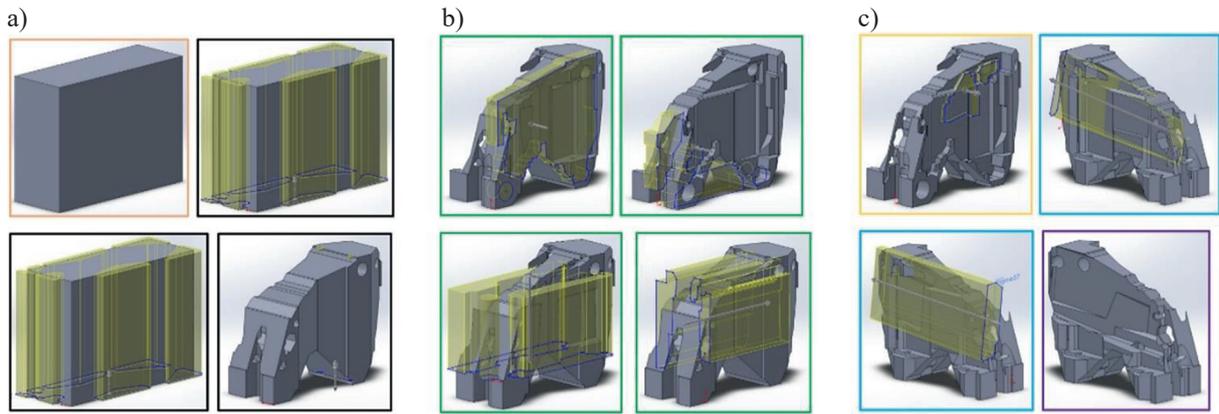
### USING A CAD SYSTEM TO RECREATE A 3D MODEL

The nodes responsible for simplified sketches are loaded by means of a macro written in VB.NET. Initially, an extrude operation is performed this only time. In this way, a cuboid with maximum dimensions of the optimized object (Figure 5a) is created.

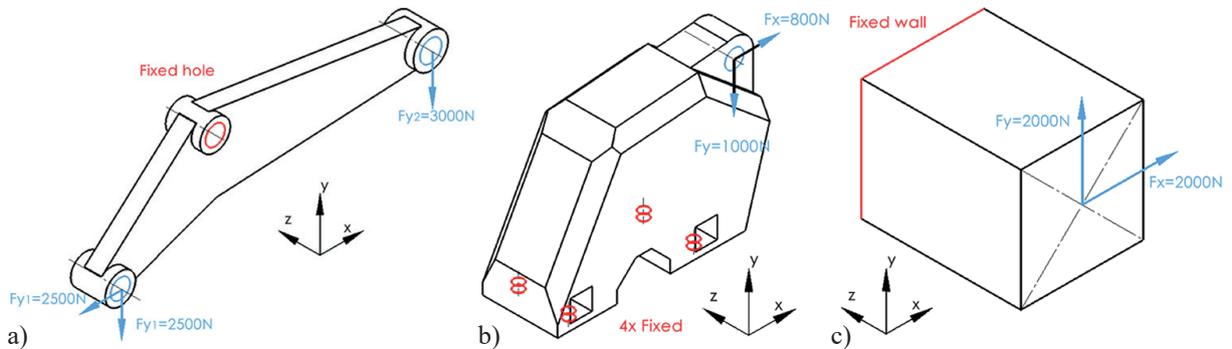
Then the operation of cutting through-all, and blind holes (Figure 5a, b) is performed. This is followed by a removal of residual material using prism and pyramid geometry (Figure 5b, c). In each of the above-mentioned definitions of holes, “cut-extrude” operations were used. If the applied operation has no significant effect on the volume of the solid, it is cancelled. Finally, the issue of pieces of material hanging in the air is resolved by removing them from the workspace. It is possible to automatically apply fillets to sharp edges after the modelling process is complete.



**Fig. 4.** Graphic diagram of creating a 3D model (macro written in VB.NET for CAD system in Solidworks)



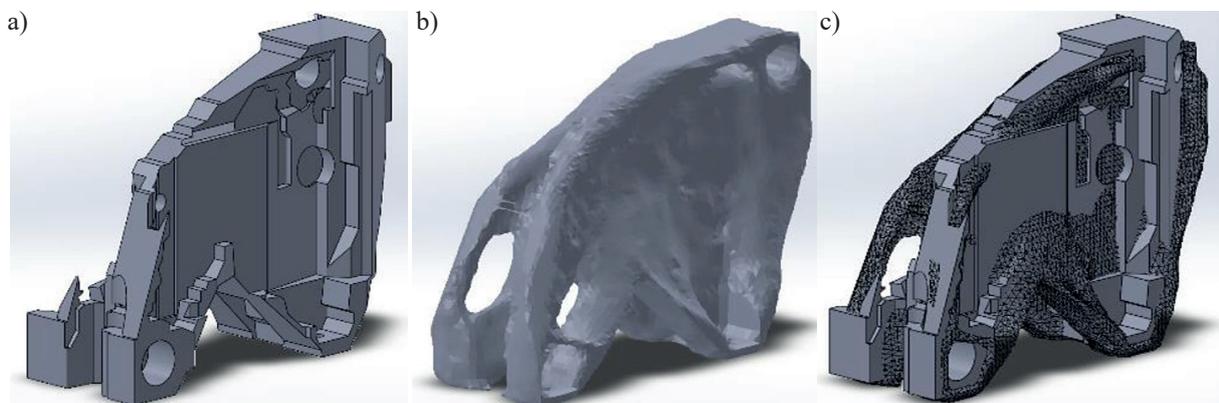
**Fig. 5.** (a) Step 3.1 - complete model creation; (orange); step 3.2 - cutting through-all holes in XYZ axes; (in black); (b) step 3.3 - cutting blind holes in XYZ axes, (green); step 3.4.1 - cutting out the material in the shape of prism geometry (yellow); (c) step 3.4.2 - cutting out the material matching pyramid template (blue); step 3.5 - removal of hanging pieces of material (purple)



**Fig. 6.** Analysed examples of topological optimization - boundary conditions (red - supports, blue - loads): (a) stabilizer, 2D example, (b) arm, 3D example (c), prism, example with multi load cases



**Fig. 7.** (a) Result of algorithm operation in CAD, (b) result exported from the optimizer in STL, and (c) comparison of both stabilizer models



**Fig. 8.** (a) Result of algorithm operation in CAD, (b) result exported from the optimizer in STL, and (c) comparison of both arm model



Fig. 9. a) Result of algorithm operation in CAD, (b) result exported from the optimizer in STL, and (c) comparison of both prism models

### EXAMPLES

The operation of the algorithm has been presented on three examples (Figure 6). The stabilizer is an example where the result is a 2D solution, the arm represents a 3D model, and a prism has been optimized using multi load cases. The stabilizer consists solely of a through-all hole cutting operation, the prism – of through-all holes, blind holes and prism-shaped holes, and the arm – of all types of holes, including prism/pyramid. A visual preview of the results is shown in Figures 7–9.

The comparison of the models was made using an operation in Solidwork called “Join” which allows Boolean operations to be performed on at least two models.

The comparison of the models was performed by means of two Boolean operations. In the first case, it was a difference between the model exported from the optimizer and the geometry transferred to CAD environment using the described algorithm. The value obtained is described as a True Positive Fraction.

Receiver operating characteristics are typically used to measure classifier effectiveness. If the case is positive and it is also classified as positive, then it counts as True Positive. If the incorrect example is classified as positive, then it is considered False Positive. However, incorrectly classified negative examples divided by “Total Negatives” result in a False Positive Fraction (FPF) [16].

In FPF, the difference between a full model resulting from an “extrude” operation and the model exported from the optimizer is defined as a negative result – “True Negative” (TN). The initially produced material is based on maximum dimensions of the optimized model in XYZ axes.

The mass factor shows how many times the volume of unnecessary external material transferred to the CAD model is greater compared to the exported results.

$$TPF = \frac{m_{org} - m_{alg}}{m_{org}} * 100\% \quad (1)$$

$$FPF = \frac{m_{alg} - m_{org}}{TN} * 100\% \quad (2)$$

$$m_{nn} \text{ ratio} = \frac{m_{alg} - m_{org}}{m_{org}} * 100\% \quad (3)$$

$$TN = m_{ex} - m_{org} \quad (4)$$

where: *TPF* – True Positive Fraction,  
*FPF* – False Positive Fraction,  
*m<sub>org</sub>* – mass of the model exported from the optimizer,  
*m<sub>alg</sub>* – mass of the model transferred to CAD system using an algorithm,  
*m<sub>ex</sub>* – mass of the prism with maximum external dimensions of the optimized model,  
*m<sub>nn</sub>* – redundant mass factor,  
*TN* – total negatives (*m<sub>ex</sub>* – *m<sub>org</sub>*).

Table 1. Comparison of the model exported from the optimizer with a CAD model obtained using the algorithm

Model ▶	Stabilizer	Arm	Prism
Parameter ▼			
TPF [%] (higher better)	85.9	65.7	73.2
FPF [%] (lower better)	3.5	11.6	13.0
<i>m<sub>nn</sub></i> ratio[%] (lower better)	29.4	34.4	136.9
False result [kg]	0.52	6.68	11.59
<i>m<sub>alg</sub></i> [kg]	2.06	19.45	17.78
<i>m<sub>org</sub></i> [kg]	1.79	19.43	8.46
<i>m<sub>ex</sub></i> [kg]	16.89	76.91	97.50

**Table 2.** Comparison of von Mises displacements and stresses between the model transferred to the CAD environment and the direct result from the optimizer

Case	Stabilizer	Arm	Prism	
Max displacement in Optistruct [mm]	0.467	0.004	0.023	0.023
Max displacement in algorithm [mm]	1.05	0.058	0.013	0.016
Max von Mises stresses in Optistruct [MPa]	101	18.3	37,6	37,5
Max von Mises stresses in algorithm [MPa]	283	20.6	56.9	61.4

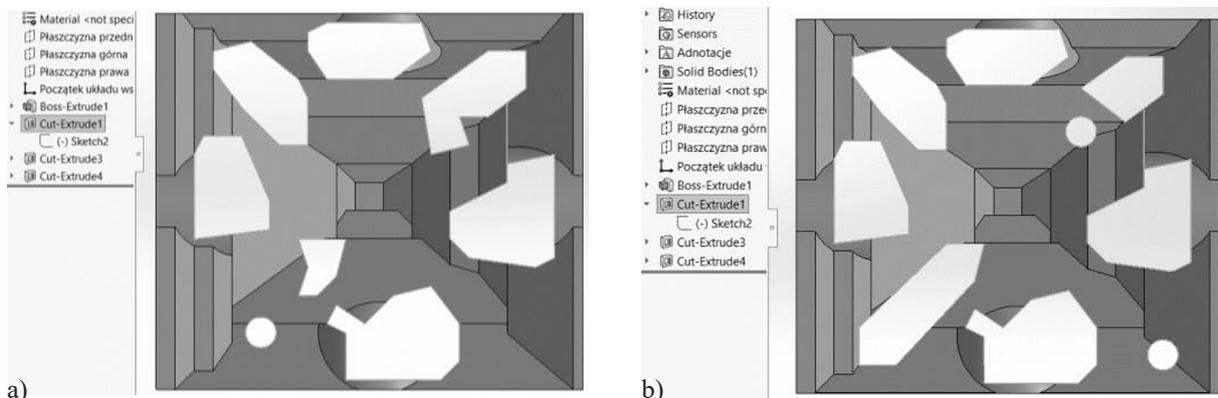
The True Positive Fraction shows how much of the model’s mass is correctly defined, and the False Positive Fraction represents the amount of material that is outside of the source model being compared.

The presented examples are characterized by TPF coefficient higher than 65%, the average being 75%, which means that the shape of the transferred solid is quite well mapped in the CAD environment. However, the FPF values do not exceed 13%, and are on average level of 9.4%, and thus a small amount of material, the CAD model, goes beyond the direct result from the optimizer.

The CAD system model was checked and compared with the direct results from the optimizer in terms of displacements and stresses in

the systems. In each case, the stresses in CAD models were higher than the results from the optimizer, on average 2.2 times. As for displacements, the stabilizer and the arm were characterized by higher values in the CAD model, on average 2.4 times, but in the case of a prism, the values were lower than the direct results from the optimizer by approximately 37%.

Table 3 shows the amount of removal material depending on the cutting operation used. Most of the material is removed in a cut through operation, an average of as much as 89% of removable space. In the third example, the prism, the amount of material removed in the CAD model was finally greater than previously

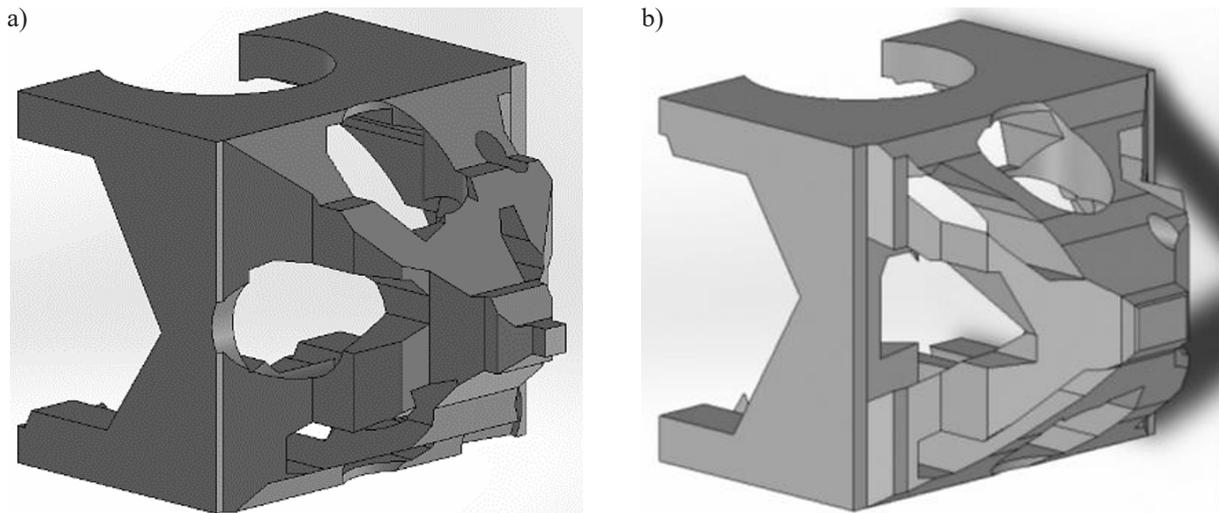


**Fig. 10.** An example of editing a sketch of a material cutting operation: (a) state before correction, (b) correction result

**Table 3.** The amount of material removed in individual hole types in the tested models

Model ▶	Stabilizer [%]		Arm [%]		Prism [%]		Average [%]	
	Va	Ve	Va	Ve	Va	Ve	Va	Ve
Parameter ▶ Holes type ▼								
Through-all	39	97	18	66	36	105	38	89
Blind	6	3	15	25	9	1	10	13
Prism	-	-	<1	<1	<1	<1	-	-
Pyramid	-	-	7	9	-	-	7	9
Average/Sum	23	100	11	100	23	115	18	111

Va - average cut volume in relation to the currently remaining volume of the model being cut.  
Ve - volume cut in relation to the entire defined “removable” volume at the start.



**Fig. 11.** Practical application of the presented method - direct editing of the 3D model in CAD system: (a) fully automated result transfer to CAD environment, and (b) the model after partial editing by the user

defined in the algorithm by approximately 15%. In the remaining cases, the same amount of mass was removed from the CAD model as initially specified in the algorithm.

### PRACTICAL APPLICATION OF THE METHOD – DIRECT EDITION OF 3D MODEL IN CAD SYSTEM

The model obtained by the algorithm can be easily and conveniently modified. First of all, the operation tree is available, which allows you to edit the already existing material cutting operations. You can lengthen or shorten an already existing hole, change its shape (Figure 10), or even delete the operation if it does not correspond to the engineer's vision. In addition, new operations for both removing and adding material can be added to obtain the shape desired by the user. You can see the results of the editing on Figure 11b).

### CONCLUSIONS

The article presents a method of transferring topological optimisation results directly to CAD system. True Positive Fraction TPF ratio averaged 75%, with an average False Negative Fraction - FNF value of 9.4%, which means that the shape of the transferred models largely corresponds to direct optimizer results. The stresses and displacements are usually higher in the models transferred to CAD, on average 2.2 times and 2.4 times respectively, but they are still

within acceptable values; in the case of stresses they are below the yield strength of steel.

The obtained models can be easily and conveniently edited by adding new operations for removing and adding material or by editing already existing operations in CAD operation tree. This means that the tedious process of recreating the model from scratch can be skipped. The user gets a tool that speeds up the process of transferring topological optimization results to CAD environment and allows the model to be adapted to appropriate technological process.

The obtained results prove that the discussed method should be further developed so that the model transferred to CAD environment would have the best possible shape to match the direct model from the optimizer. This means a high TPF value, closer to 100%, and the lowest possible FPF value, tending to 0%.

### REFERENCES

1. Lin C.Y., Chao L.S. Automated image interpretation for integrated topology and shape optimization. *Structural and Multidisciplinary Optimization*. 2000; 20: 125–137.
2. Lin C.Y., Chou Y.H. Automated structural optimization system for integrated topology and shape optimization. *Journal of the Chinese Institute of Engineers*. 2008; 31: 745–756.
3. Chacón J.M, Bellido J.C, Donoso A. Integration of topology optimized designs into CAD/CAM via an IGES translator. 2014; 50: 1115–1125.
4. Tang P.S., Chang K.H. Integration of topology and shape optimization for design of structural compo-

- nents. *Structural and Multidisciplinary Optimization*. 2001; 22: 65–82.
5. Hsu M.H., Hsu Y.L. Interpreting three-dimensional structural topology optimization results. *Computers and Structures*. 2005; 83: 327–337.
  6. Larsen S., Jensen C.G. Converting Topology optimization Results into Parametric CAD Models. *Computer-Aided Design and Applications*. 2009; 6: 407–418.
  7. Cuillière J.C., Franço V., Nana A. Automatic Construction of structural CAD models from 3D topology optimization. *Computer-Aided Design and Applications*. 2017; 15: 107–112.
  8. Yin G., Xiao X., Cirak F. Topologically robust CAD model generation of structural optimization. *Computer methods in applied mechanics and engineering*. 2020; 369: 113102.
  9. Kresslein J., Haghghi P., Park J., Satchit R., Sutradhar A., Shah J.J. Automated cross-sectional shape recovery of 3D branching structures from point cloud. *Journal of Computational Design and Engineering*. 2018; 5: 368–378.
  10. Shabani B., Dukovski V. Integration of Reverse Engineering and Topology Optimization with Additive Manufacturing. 2022; 19(1): 164–175.
  11. <https://www.tiobe.com/tiobe-index/> [accessed 06 Nov. 2021].
  12. <https://insights.stackoverflow.com/trends?tags=python%2Cc%2Cc%2B%2B%2Cvb.net%2Cjava%2Cjavascript> [accessed 06 Nov. 2021].
  13. <https://pypl.github.io/PYPL.html> [accessed 06 Nov. 2021].
  14. Guo L., Zhou M., Lu Y., Yang T., Yang F. A hybrid 3D feature recognition method based on rule and graph, *Computer Integrated Manufacturing*, 2021; 34: 257–281.
  15. Vasantha G., Purves D., Quigley J., Corney J., Sherlock A., Randika G. Common design structures and substitutable feature discovery in CAD databases, *Advanced Engineering Informatics*. 2021; 48: 101261.
  16. Fawcett T. An Introduction to ROC analysis. *Pattern Recognition Letters*. 2005; 27: 861–874.