23rd International Conference
ENGINEERING MECHANICS 2017
Svratka, Czech Republic, 15 – 18 May 2017

FAILURE INDEX BASED TOPOLOGY OPTIMIZATION FOR MULTIPLE PROPERTIES

F. Löffelmann*

Abstract: *Aim of the presented work is to introduce a method which is being coded. It works in conjunction with an open-source finite element solver CalculiX. One of the topology optimization methods - The Bi-directional Evolutionary Structural Optimization method (BESO) was chosen due to its simplicity, which is used in the way of python code standing above FE solver. Two test examples are presented. First demonstrates ability to achieve results similar to compliance based topology optimization on short beam leading to solid-void structure. Second example shows possibility to switch among different materials with regard to their allowable stresses.*

Keywords: Structural optimization, Heuristic, BESO, Solid-void, Multiple material.

1. Introduction

Topology optimization searches for convenient material distribution of product in the design space. In recent years it is increasingly used thanks to increase in computational power and increasing number of software integrating FEM and topology optimization. Out of complex optimization packages there are several optimizers freely available (non-trial versions), but only a few with open-source codes, mostly scientific explanatory papers, e.g. Sigmund (2001), Zuo et al. (2015), or a master thesis Denk (2016). A presented python code is available on https://github.com/fandaL/beso and it works as an optimizer which currently uses the open-source FE solver CalculiX and a user defined input file for a common FE analysis. The BESO method was initially taken from Huang (2010), but in the current state it is significantly modified. Contrary to most of topology optimizers, which are compliance based or stress based (e.g. Querin, 1997) and modifies element stiffness matrices, this code is close to stress based methods since it is based directly on failure indices and modifies element materials which both are listed by a user. Similar tasks were solved also by Ramani (2011).

2. Methods

Description follows schema in Fig. 1. At the beginning a user defines common FE analysis and sets optimization parameters, from which most important are domain parameters (lists of switching materials, effective densities, thicknesses, failure criteria; lists must be ordered from weakest to strongest material), type of filter and filter radius, and mass goal ratio.

Several operations can be done only ones before iterating. It is importing a mesh for user defined domains to be optimized or only included in monitoring and filter range. Another operation is computing volumes (resp. areas of shells) to evaluate mass during iteration. Volumes together with centers of gravity and finding near elements are used for filtering.

Basic measure of efficiency of each element is in an actual state defined as its sensitivity number

$$\alpha = \frac{FI_e}{\rho_e}\ , \ \ FI_e = \frac{\sigma_{e,evaluated}}{\sigma_{e,allowable}} \qquad (1)$$

Higher element failure index FI_e means that element is used closer to its failure capacity and division by effective element density ρ_e is used to prefer lighter (cheaper) material. σ_e is element stress.

* Ing. František Löffelmann: Institute of Aerospace Engineering, Brno University of Technology, Technická 2896/2, 616 69 Brno, CZ, Frantisek.Loffelmann@vutbr.cz

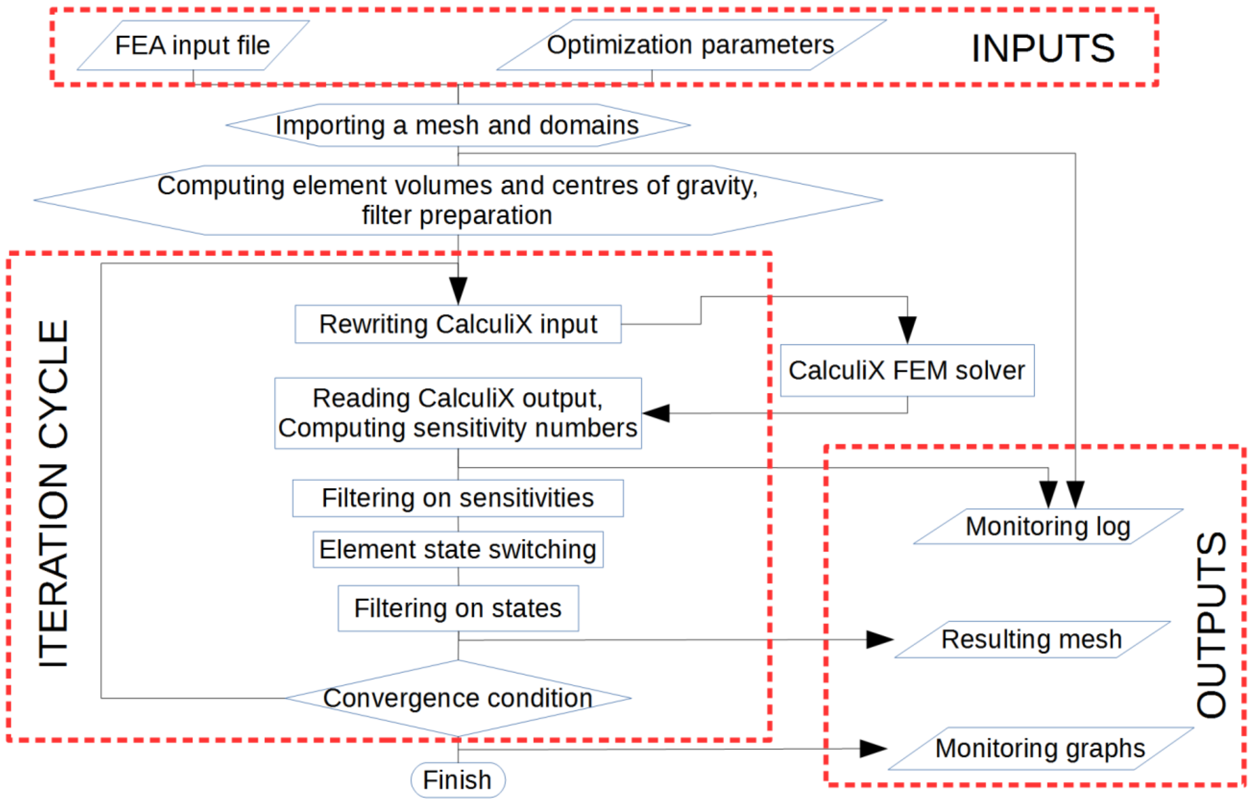


Fig. 1: Simplified flowchart of the program.

Very important section is for element switching. In each iteration prescribed mass is switched up (to stronger state) defined by Mass Addition Ratio *AR*, and in the same iteration mass defined by Mass Removal Ratio *RR* is switched down (to weaker state). Difference between them yields goal decrease in total mass from the previous iteration. Removing is limited by achieving the mass goal, or by exceeding allowable number of violated elements which have failure ($FI_e \geq 1$). Violated elements are forced to switch up and their mass difference is added to the goal mass of the current iteration. The rest of elements is ordered according to their sensitivity numbers (eq. 1). Switching up is done from elements with highest sensitivity number, then switching down from the lowest (least effective elements).

When failure occurs on more than allowable number of elements, decaying is activated which means that Mass Addition Ratio *AR* exponentially decreases between iterations. Speed of decaying is driven by exponent *k* in the equation

$$AR_i = AR_0 \cdot e^{k\cdot(i-i_v)}\ ,\ k < 0 \qquad (2)$$

Where AR_0 is the Mass Addition Ratio used at the beginning, i is the actual iteration number, and i_v is number of the iteration in which tolerance in failed elements was exceeded. Until more failing elements are present, Mass Removing Ratio equals actual AR_i which assures that mass is not removed. Using this approach is an artificial way how to stabilize results from continuing switching some elements up and others down.

Criterion for terminating iterations is given by mean *FI* weighted by element mass which should remain in prescribed tolerance for the last 5 iterations. Iterating can be also terminated by the maximum number of iterations.

3. Examples

Solid-void properties

A short cantilevered beam modeled in 2D was picked since it can be compared with typical results of usually used compliance based methods. The initial model is shown in Fig. 2. Material was isotropic linear elastic with a modulus E = 70 GPa, Poisson ration μ = 0.3 and chosen effective density 1, allowable von Mises stress 320 MPa. Void material properties were set to low modulus E = 70 Pa, μ = 0.3, low effective density 10^{-6}, but high stress limit 320×10^6 MPa setting void material far from failure.

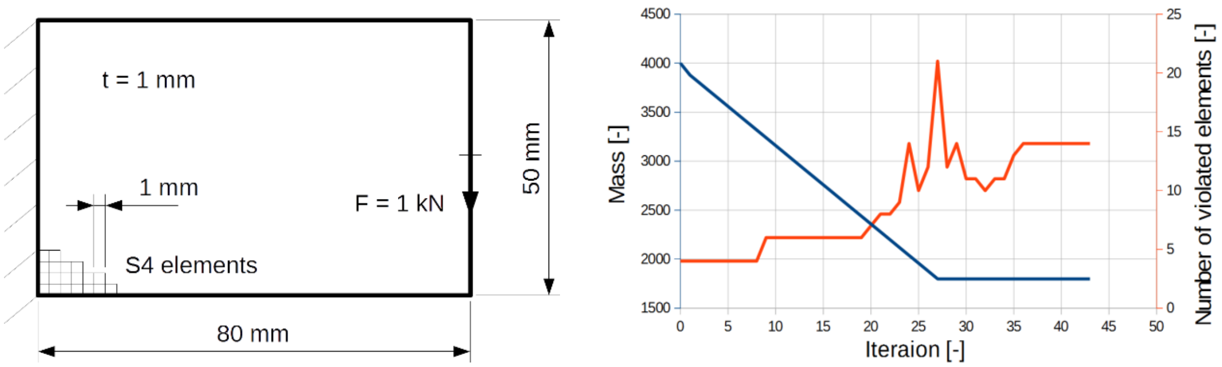
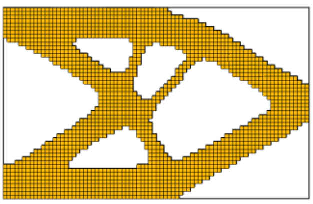
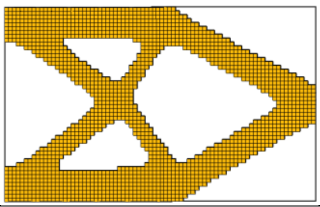
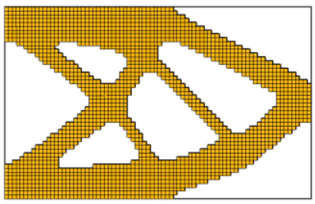


Fig. 2: Short beam schema (left), Mass and number of failed elements (right) for AR = 0.01 and RR = 0.03.

Results for 3 different settings are in Tab. 1. Mass Addition Ratio *AR* = 0.01 and Mass Removal Ratio *RR* = 0.03 means that in each iteration 2 % of initial mass was removed. Resulting mesh for these settings corresponds well with a comparison of this task between BESO and SIMP method described by Huang (2007). In this case exactly the same filter scheme as in the literature for BESO method was used with same filter range 3 mm, which is described there and also in the book Huang (2010). Tolerance of failed elements was 10 over initial failed elements, i.e. 14 elements.

Tab. 1: Results for different Mass Addition Ratio and Mass Removal Ratio.

AR RR	0.005 0.015	0.01 0.03	0.015 0.045
Iterations	63	43	31
Failing elements	15	14	12
Resulting mass	1899	1799	1899
Resulting mesh			

Multiple properties

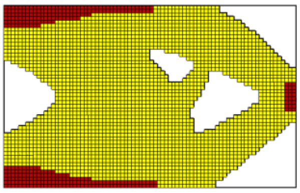
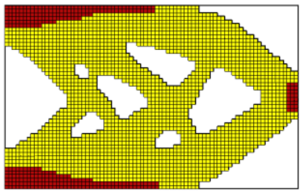
In the next example same geometry as in Fig. 2-left was used. A difference was in using multiple materials. First material was the same (Al alloy in Tab. 2). Second material had lower properties (with effective density as a ratio of magnesium and aluminum density 1.7 / 2.8 = 0.607, void material was set in the similar way as previously. Mass Addition Ratio AR = 0.01 and Mass Removal Ratio *RR* = 0.03 were taken as relative values to actual mass (not related to initial mass as in the first example). In case of multiple properties (states) it is not possible to use simple filtering of sensitivities as previously. Instead open-close filter working on element states was used with radius 1.5 mm. This morphology based filter works on principles used in topology optimization by Sigmund (2007).

Tab. 2: Used material properties.

		Al alloy	Mg alloy	Void material
E	[GPa]	70	45	45×10^{6}
μ	[-]	0.3	0.3	0.3
Effective density	[-]	1	0.607	0.607×10^{-6}
Allowable von Mises stress	[MPa]	320	150	150×10^{-6}

In the first run model finished on higher mass then in case of 1 material (stronger one) in the previous example because of violated elements. Higher mass was caused by removing least effectively used material which was also stronger one (farther from allowable stress) so that softer material stayed also less effectively used. Second run was a continuation of the first one where elements, which ended previously in the stronger state, were frozen by excluding from optimization domain. This process led to significantly lower mass but also more elements with exceeded allowable stress (Tab. 3).

Tab. 3: Results for the model with 2 solid materials and void material.

	First run	Continuation	
Iterations	40	19	
Failing elements	7	17	
Resulting mass	2042	1433	
Resulting mesh			Al alloy Mg alloy

4. Conclusions

The method, which was described, was tested on simple examples. Short beam example lead to the result corresponding with BESO and SIMP method for a “solid” or “void” material, although some level of result dependency on input parameters is obvious from Tab. 2. Second example tested switching among different properties from a prescribed list given by 2 solid materials and void material. Manual freezing of elements in stronger state was needed to achieve lower mass which was expected by adding softer material in compare to the first example. Programmed optimization method can be used on more complex problems but with aware that it may automatically finish too early and results can be dependent on parameter settings.

Acknowledgement

The research leading to these results has received funding from the MEYS under the National Sustainability Programme I (Project LO1202).

References

Denk, M. (2016) Entwicklung einer Applikation für eine automatisierte multiphysikalische Topologieoptimierung und Geometrierückführung. Hochschule für angewandte Wissenschaften München.

Huang, X. and Xie, Y.M. (2007) Convergent and mesh-independent solutions for the bi-directional evolutionary structural optimization method, Finite Elements in Analysis and Design, 43, 14, pp. 1039-1049.

Huang, X. and Xie, Y.M. (2010) Evolutionary Topology Optimization of Continuum Structures: Methods and Applications. John Wiley & Sons. Hoboken, NJ, USA.

Ramani, A. (2011) Multi-material topology optimization with strength constraints. Struct Multidisc Optim 43, pp. 597-615. DOI 10.1007/s00158-010-0581-z.

Sigmund, O. (2001) A 99 line topology optimization code written in Matlab. Structural and Multidisciplinary Optimization, 21, pp. 120–127. DOI: 10.1007/s001580050176.

Sigmund, O. (2007) Morphology-based black and white filters for topology optimization. Struct Multidisc Optim 33, pp. 401-424. DOI: 10.1007/s00158-006-0087-x.

Querin, O.M. (1997) Evolutionary Structural Optimisation: Stress Based Formulation and Implementation. PhD. Dissertation, Department of Aeronautical Engineering, University of Sydney.

Zuo, H.Z. and Xie, Y.M. (2015) A simple and compact Python code for complex 3D topology optimization. Advances in Engineering Software 85, pp. 1-11.