Structural and Multidisciplinary Optimization
https://doi.org/10.1007/s00158-021-03050-7

REVIEW PAPER

A comprehensive review of educational articles on structural and multidisciplinary optimization

Chao Wang[1] · Zhi Zhao[1] · Ming Zhou[2] · Ole Sigmund[3] · Xiaojia Shelly Zhang[1,4]

Received: 18 June 2021 / Revised: 20 July 2021 / Accepted: 6 August 2021

Abstract
Ever since the publication of the 99-line topology optimization MATLAB code (`top99`) by Sigmund in 2001, educational articles have emerged as a popular category of contributions within the structural and multidisciplinary optimization (SMO) community. The number of educational papers in the field of SMO has been growing rapidly in recent years. Some educational contributions have made a tremendous impact on both research and education. For example, `top99` (Sigmund in Struct Multidisc Optim 21(2):120–127, 2001) has been downloaded over 13,000 times and cited over 2000 times in Google Scholar. In this paper, we attempt to provide a systematic and comprehensive review of educational articles and codes in SMO, including topology, sizing, and shape optimization and building blocks. We first assess the papers according to the adopted methods, which include density-based, level-set, ground structure, and more. We then provide comparisons and evaluations on the codes from several key aspects, including techniques, efficiency, usability, readability, environment, and compatibility. In addition, we conduct numerical experiments on the reviewed codes using the benchmark cantilever beam example to provide feedback on the overall user experience. With a systematic review and comparison, this paper aims to offer insights on the educational values and practicality for employing these codes. We try to provide not only guidance for beginners to approach various optimization methods, but also a dictionary to direct readers to effectively target the relevant codes and building blocks based on their demands. Finally, based on the findings in this review paper, we provide some perspectives and recommendations for future educational contributions.

Keywords Educational codes · Educational contributions · Topology optimization · Sizing optimization · Shape optimization · Building blocks

1 Introduction

Structural and multidisciplinary optimization (SMO) has received considerable attention over the past decades spanning a wide range of disciplines, including structural mechanics, fluids, material science, acoustics, biomedical, optics, and more. SMO methods are generally classified into three categories: topology, sizing, and shape optimization. Topology optimization aims to optimize both geometric features and connectivity within a design domain. Sizing optimization refers to optimizing the structural dimensions such as cross-sectional areas of truss members or the thickness distribution of a shell structure. Shape optimization attempts to optimize the contour of structural boundaries without changing the connectivity of structural members. Note that the boundaries between the above categories are generally fuzzy, which often depend on selections of the finite element (FE) models. For example, thickness sizing

Responsible Editor: Gengdong Cheng

Special Issue dedicated to Dr. Raphael T. Haftka

✉ Xiaojia Shelly Zhang
zhangxs@illinois.edu

[1] Department of Civil and Environmental Engineering, University of Illinois at Urbana Champaign, 205 North Mathews Ave, Urbana, IL 61801, USA

[2] Altair Engineering, Altair Engineering, Irvine, CA, USA

[3] Department of Mechanical Engineering, Solid Mechanics, Technical University of Denmark, 2800 Lyngby, Denmark

[4] Department of Mechanical Science and Engineering, University of Illinois Urbana-Champaign, 1206 W. Green St., Urbana, IL 61801, USA

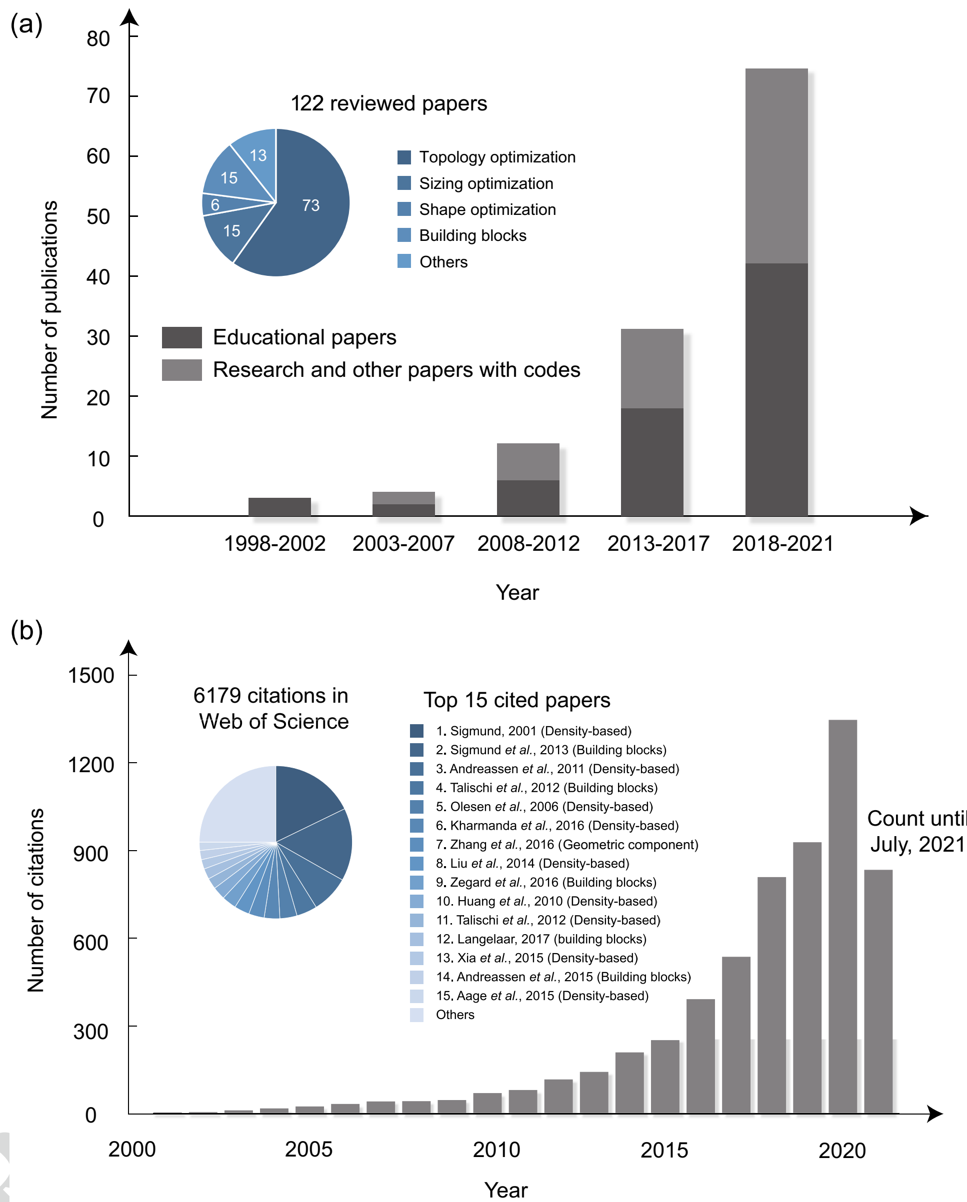


Fig. 1 Statistics of 122 reviewed papers (including educational, research, and other paper types): **a** the increasing number of publications and general categories of SMO methods; **b** the increasing number of citations and the top 15 cited papers

optimization of a 2D shell structure has to be achieved by shape optimization when the structure is modeled with 3D solid elements. Generally speaking, topology contains all three categories as it defines structural connectivity, shape, and size in a unified framework.

With the vast research developments in the SMO field, educational articles have emerged as a popular category of contributions within the community since the publication of the 99-line topology optimization MATLAB code (Sigmund 2001). In this review, we identified a total of 122 papers with educational components, comprised of mostly educational papers but also some research, review, and forum discussion papers with a strong focus on codes. Among them, many papers aim to provide standard or specialized educational codes that solve various types of SMO problems, with a clear objective to facilitate beginners and researchers to learn detailed implementation of an established method, e.g., the 99/88-line codes (Sigmund 2001; Andreassen et al. 2011) and `PolyTop` code (Talischi et al. 2012b), or to introduce a new method to the community with hands-on experience, e.g., moving morphable components (MMC) method (Guo et al. 2014; Zhang et al. 2016b). Others are articles with educational purposes aiming at explaining or discussing fundamental and critical concepts for topology optimization problems, for instance, educational papers by Stolpe (2010) and Klarbring (2015). We categorized the collected papers into several groups and summarized them in Fig. 1a. From the figure, we observe an increasing trend in the number of educational papers and research (and other) papers with code focus

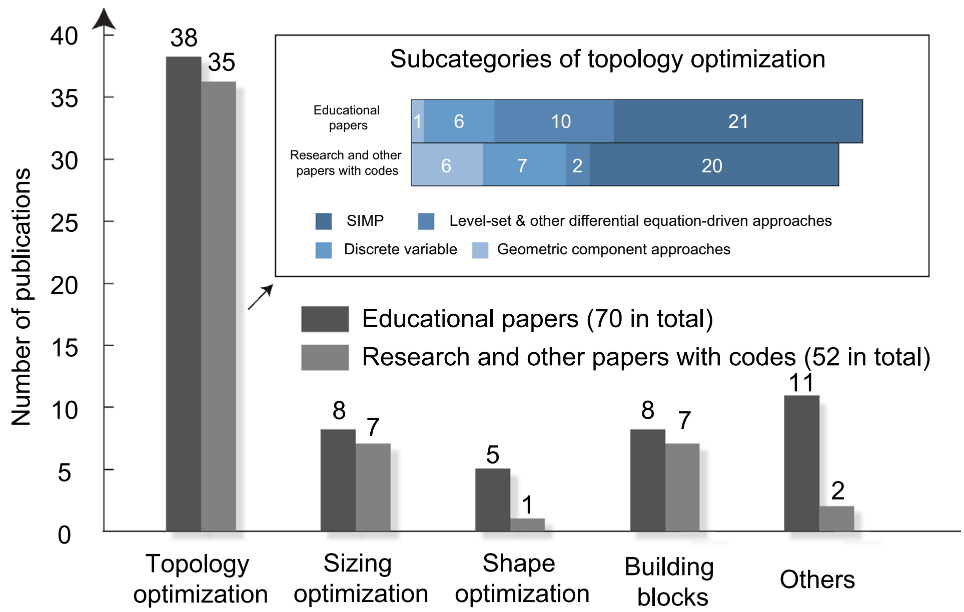


Fig. 2 Statistics of the reviewed papers (including educational, research, and other paper types) categorized by different adopted methods

over the years, with the topology optimization category constituting the largest portion. These papers with educational contributions have created an enormous impact on the SMO field. Figure 1b shows the number of total citations of those 122 papers received every year together with a list of the top 15 most cited papers. The growing trend in the number of citations over the years demonstrates the tremendous impact and benefits those articles with educational values have brought to the SMO community.

In light of the enormous influence of these articles on both education and research in SMO, this study aims to conduct a systematic and comprehensive review of educational contributions on SMO, with a particular focus on the coding and computational aspects. To that end, we collected articles that are labeled as educational papers and other paper types that provide codes (or Apps) via electronic supplementary material (ESM), appendices, or other platforms. Figure 2 shows the statistics of the reviewed papers in each method. The codes contained in those collected articles are reviewed and evaluated. The purpose of this work is to offer insights on educational values and practicality for employing these codes. With the systematic review and evaluation, this paper can serve not only as a guide for beginners to approach different optimization methods but also a dictionary for researchers targeting specialized problems or in need of building blocks based on their demands.

To provide a thorough review of the collected SMO codes, we evaluate them based on several key dimensions, including techniques related to FE analysis, optimization, and programming aspect (e.g., efficiency and parallelization, etc.); code environment; usability; readability; and compactness. For codes written in the MATLAB environment, the usability was tested using MATLAB R2020a, and we report the compatibility with GNU Octave (6.2.0), which is a popular open-source alternative to MATLAB. All the codes are categorized and compiled into tables following the same order as discussed in the text, which may serve as a dictionary for readers to quickly locate a particular code and corresponding reference. For this purpose, the DOI (Digital Object Identifier) information of each paper is also included in the tables. To further facilitate identification, we include the code names either officially proposed by the authors or from the code function names in the tables (except for those without explicit name information). We also provide a summary column collecting the main features for each code. Finally, we conduct numerical experiments on the codes that solve the classic compliance minimization problem using the benchmark cantilever beam example. All codes were run "as is" with default settings reflecting direct overall user experiences.

The reviewed papers are organized into categories in the SMO field, i.e., topology optimization, sizing optimization, shape optimization, building blocks, and educational papers without codes, as shown in Fig. 3. As the first category, topology optimization approaches are further categorized into density-based methods (Bendsøe 1989; Zhou and Rozvany 1991; Bendsøe and Sigmund 1999; Xie and Steven 1993), level-set (Osher and Sethian 1988; Sethian 1999; Allaire et al. 2002; Wang et al. 2003) and other differential equation-driven approaches (Eschenauer et al. 1994; Sokolowski and Zochowski 1999; Wallin et al. 2012; Wang and Zhou 2004; Burger and Stainko 2006), and geometric component approaches (Bai and Zuo 2020; Zhao et al. 2021; Zhang et al. 2016b). In the density-based methods, the optimization is established based on elements or nodes. According to the format of design variables, density-based

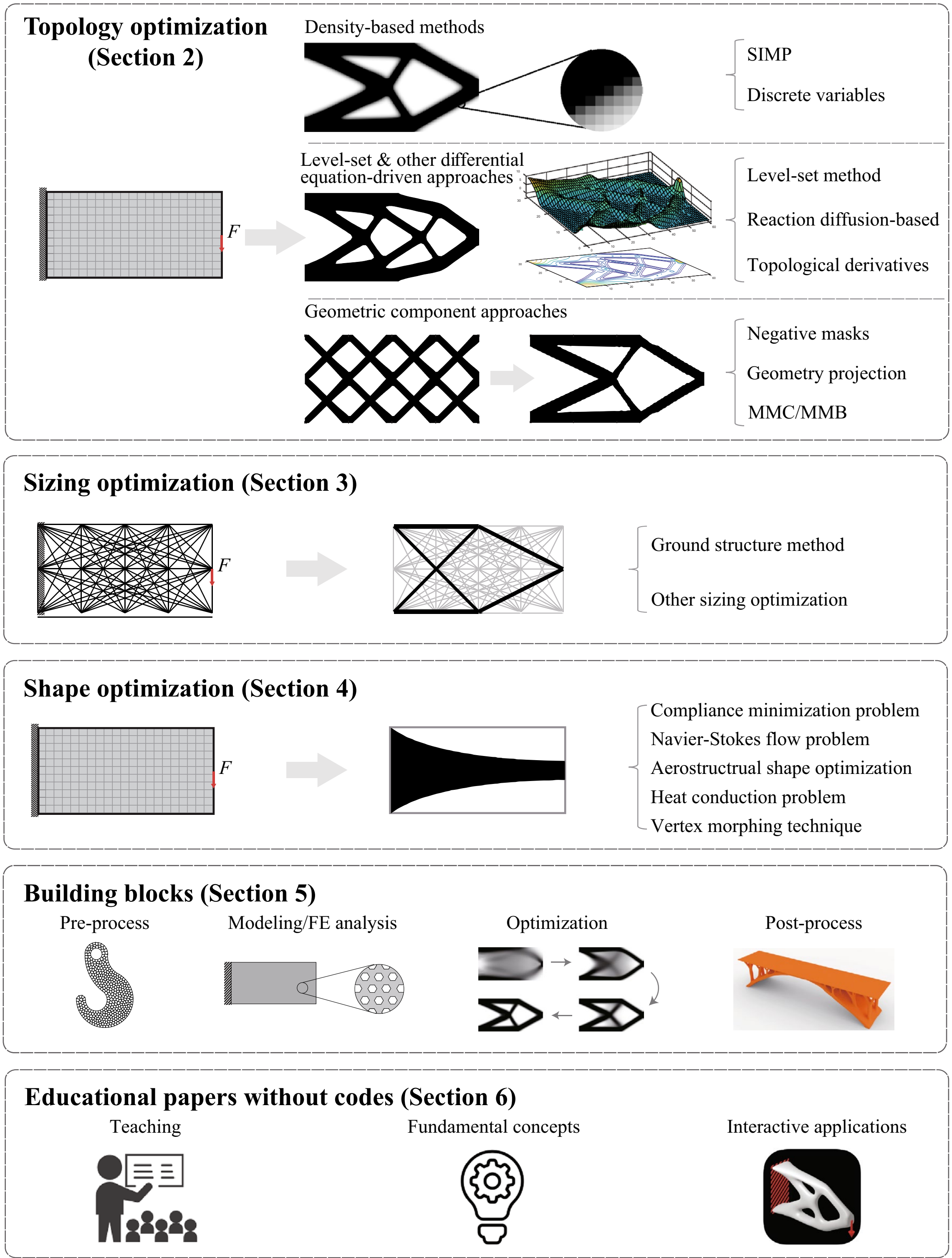


Fig. 3 Category and sub-category of reviewed papers in this study

methods are divided into Solid Isotropic Material with Penalization (SIMP) (Bendsøe 1989; Zhou and Rozvany 1991; Bendsøe and Sigmund 1999) and discrete variable approaches, where the former utilizes continuous density variables (which continuously vary between 0 and 1 with penalization of the intermediate values), while the latter employs discrete density variables (which take values of either 0 or 1), such as the ESO (evolutionary structural optimization) method (Xie and Steven 1993). The level-set and other differential equation-driven approaches include the classical level-set methods, which use the level-set function to implicitly describe the boundary of different phases, and other methods making use of various differential equations such as reaction-diffusion-based approaches and topological derivative approaches. The geometry component approaches include the method employing negative masks, geometry projection (Zhang et al. 2016a), and MMC and moving morphable bars (MMB) methods (Guo et al. 2014; Zhang et al. 2016b; Zhao et al. 2021). For sizing optimization, reviewed articles are further categorized into ground structure method and others. For the shape optimization category, the work is evaluated based on different topics of problems: compliance minimization, Stokes flow, aerostructural shape optimization, and heat conduction problems. For the building block category, we review papers that specifically discuss one (or more) building block(s) of an SMO procedure, such as mesh generation, FE analysis, design update scheme, and post-processing. For the educational papers without codes, we review papers with educational values related to teaching, fundamental concepts, and interactive applications.

It is worth mentioning that this paper will focus on the review of educational contributions of various SMO methods and keep the discussions on other aspects (e.g., technical details, comparisons, and derivations) to a minimal extent. For detailed overviews of these other aspects of various SMO methods, we refer the readers to other review articles. For example, see Sigmund and Maute (2013) for an overview and comparison of different approaches in topology optimization and perspectives on the trend and future directions; Xia et al. (2018) for an introduction of the evolutionary approaches; van Dijk et al. (2013) for a comprehensive review about the level-set approaches including level-set function parameterization, geometry mapping, mechanical modeling, and update procedure; Deaton and Grandhi (2014) for a review of the application of topology optimization methods in multiple disciplines; Rozvany (2009) for review work focusing on numerical methods reaching the stage of application in industrial software; Wein et al. (2020) for the category of methods solving structural optimization problems termed feature-mapping methods; Stolpe (2016) for articles in truss optimization with deterministic optimization methods and meta heuristics.

The remainder of the paper is organized as follows: In Sect. 2, we review the educational contributions on topology optimization. The educational papers (and other types of papers) with codes for sizing and ground structure approaches are discussed in Sect. 3, and the shape optimization is reviewed in Sect. 4. In Sects. 2–4, basic parametrization and/or formulation for different methods are presented. Thorough evaluations on the codes from several aspects are provided. Section 5 reviews building block codes for various SMO methods. Papers that focus on educational values other than codes are reviewed in Sect. 6. In Sect. 7, numerical experiments using standard codes based on default parameters are conducted to provide a snap-shot of overall user experiences. Finally, conclusions and perspectives are drawn in Sect. 8.

2 Topology optimization

The general topology optimization problem aims at finding the material distribution within a prescribed design domain that minimizes the objective function subject to a set of constraints. This section reviews educational contributions (i.e., 38 educational papers and 35 other types of papers that provide codes) in the field of topology optimization, which is categorized into density-based methods, level-set and other differential equation-driven methods, and methods using geometric components/bars.

2.1 Density-based methods

In the density-based method, the design domain is discretized by a mesh of finite elements, and the density for each element is optimized. This section first provides an overview of the basic formulation. We then review a total of 27 educational papers and 27 research and other papers that use the density-based methods.

The basic optimization formulation for density-based method is as follows (Bendsøe and Sigmund 2013):

$$
\begin{aligned}
&\min_{\rho} : J(\boldsymbol{\rho}, \mathbf{u}(\boldsymbol{\rho})) \\
&\text{s.t.} : \sum_{e=1}^{N_e} v_e \rho_e - V_0 \le 0 \\
&g_i(\boldsymbol{\rho}, \mathbf{u}(\boldsymbol{\rho})) \le 0 \quad i = 1, \dots, m \\
&\boldsymbol{\rho}_e \in \{0, 1\}, \quad e = 1, \dots, N_e \\
&\mathbf{Ku}(\rho) = \mathbf{F},
\end{aligned}
\tag{1}
$$

where N_e is the number of elements, ρ_e is the discrete density variable which can take the value of 0 (representing void) or 1 (representing solid), $J(\boldsymbol{\rho}, \mathbf{u}(\boldsymbol{\rho}))$ is the objective function (e.g., compliance), $\sum_{e=1}^{N_e} v_e \rho_e - V_0 \le 0$ represents the volume constraint, $g_i(\boldsymbol{\rho}, \mathbf{u}(\boldsymbol{\rho}))$, $i = 1, \dots, m$ are m other constraints, such as stress, buckling, symmetry, or maximum member size constraints, and $\mathbf{Ku}(\rho) = \mathbf{F}$ is the state equation that ensures the global equilibrium (where we consider linear elasticity for demonstration).

For problems with a large number of design variables, the discrete nature of the optimization problem makes it computationally intractable (Sigmund 2011). To enable the use of efficient gradient-based optimization algorithms, a

Journal : Large 158 | Article No : 3050 | Pages : 54 | MS Code : 3050 | Dispatch : 28-9-2021

continuous parameterization of design variables $\rho_e \in (0,1]$ is introduced together with a material interpolation scheme, which penalizes intermediate density values. A common material interpolation scheme is the SIMP method (Bendsøe 1989; Zhou and Rozvany 1991; Bendsøe and Sigmund 1999), in which the relationship between the elastic modulus and the element density is defined as,

$$E(\rho_e) = \rho_e^p E_0, \quad p \geq 1, \tag{2}$$

with p being the penalization parameter and E_0 being the Young's modulus of the solid material. This original SIMP has in later codes been substituted with

$$E(\rho_e) = E_{min} + \rho_e^p (E_0 - E_{min}), \quad \rho_e \in [0,1], \tag{3}$$

where E_{min} is the stiffness of the void material (which is nonzero to avoid singularity). The use of the modified SIMP can allow for p-independent control of the "void" stiffness.

The remaining subsections focus on educational contributions to the continuous and discrete density-based topology optimization, including standard density-based (SIMP) codes, codes for solving specialized problems, and discrete variable codes.

2.1.1 Standard density-based (SIMP) codes

A number of papers provide codes to solve the standard compliance topology optimization problem using the SIMP material interpolation scheme. The collected contributions in this category include 8 educational papers and 7 research papers, as summarized in Table 1, which follows the same order as discussed below. In 2001, Sigmund (2001) published the first educational code (99-line topology optimization code, referred to as `top99`) in MATLAB, which handles two-dimensional (2D) standard compliance minimization problems. A clear code structure and sequential implementations (and presentation) of building blocks are employed to facilitate the understanding of the entire topology optimization process, making it an excellent educational reference for students and newcomers of the field. The efficiency of the code was later improved by an 88-line MATLAB code (i.e., `top88`) (Andreassen et al. 2011), where the nested "`for`" loops in `top99` are vectorized. In addition, extensions of alternative filtering types are presented in this article. To facilitate the use of topology optimization codes for arbitrary design domains, Talischi et al. (2012b) published a MATLAB code (referred to as `PolyTop`) employing unstructured polygonal meshes (Talischi et al. 2012a) in topology optimization. `PolyTop` decouples the general FE analysis routine and optimization formulation, promoting the versatility to accommodate different formulations.

In terms of three-dimensional (3D) topology optimization problems, several codes were developed based on the standard 2D codes. Liu and Tovar (2014) introduced a MATLAB code (i.e., `top3D`), which is built upon `top88` (Andreassen et al. 2011), to handle 3D problems. An iterative solver using the built-in MATLAB function "`pcg`" is discussed in this paper to improve the efficiency of solving large-scale FE analysis. Partially based on the `top99` (Sigmund 2001) and `top88` (Andreassen et al. 2011), Lagaros et al. (2019) developed a 3D density-based topology optimization framework written in C# language, which is integrated with SAP2000 through an open application programming interface. Other than employing the most commonly used update schemes, such as OC (Optimality Criteria) method (Bendsøe and Sigmund 1995) and MMA (Method of Moving Asymptotes) (Svanberg 1987), Zeng and Ma (2020) developed 2D and 3D MATLAB codes based on the `top99` and `top88` coding structures by using a new gradient projection optimizer. Based on a coding structure similar to `PolyTop` (Talischi et al. 2012b), Chi et al. (2020) proposed a 3D topology optimization framework using polyhedral discretization, where the virtual element method (VEM) (Beirão da Veiga et al. 2013) is employed to handle arbitrary element shapes and perform structural analysis efficiently.

To improve the usability of topology optimization codes to tackle large-scale problems, computational efficiency is an imperative aspect for application. To this end, a number of codes have been developed to improve the efficiency by employing iterative solvers for FE analysis, parallel computation, machine learning, and other techniques. Amir et al. (2014) exploited the multigrid preconditioned conjugate gradients (MGCG) solver and implemented it in MATLAB to improve the efficiency of solving both 2D and 3D problems. Amir (2015) employed recycled preconditioning for 2D and 3D MGCG-based volume minimization problem with the purpose of reducing computational cost and developed a set of MATLAB codes. By employing the newest shortcuts and speed-up techniques in MATLAB, Ferrari and Sigmund (2020) developed a new generation 99-line MATLAB code, referred to as `top99neo`, and extended it to 3D to handle medium-/large-scale problems efficiently on a laptop. With respect to parallel computation, an open-source topology optimization framework based on the Portable and Extendable Toolkit for Scientific Computing (PETSc) (Balay et al. 2019) was developed by Aage et al. (2015). It is shown that the fully parallelized framework is capable of handling more than 100 million design variables. Subsequently, Zhang et al. (2021) developed an extended version, named `TopADD`, by incorporating the 2D topology optimization into the previous 3D parallel-computing framework (Aage et al. 2015). In addition, an efficient voxelizer is developed to enable arbitrary complex design domains for topology optimization. Schmidt and Schulz (2011) developed a 3D code for CUDA-enabled graphics card written in C++ language, and

Journal : Large 158	Article No : 3050	Pages : 54	MS Code : 3050	Dispatch : 28-9-2021

Table 1 Summary of standard density-based SIMP codes

Type	Name/Reference	Environment	Summary	Techniques: Topology optimization	Techniques: FE method	Techniques: Efficiency	Usability
Educational	"*top99*" Sigmund O (2001) https://doi.org/10.1007/s001580050176	MATLAB (Octave compatible)	– 2D – Compliance minimization – First educational paper with open-source codes	– Sensitivity filter – SIMP interpolation – OC update scheme	– Linear elasticity – Direct solver – Quadrilateral discretization	– Use of "*sparse*" function for solver – Precompute analytical element stiffness matrix – Label: Loop-based	– Code available at www.topopt.dtu.dk – Ready to use – Clear code structure and implementation to facilitate understanding
Educational	"*top88*" Andreassen E, Clausen A, Schevenels M, Lazarov BS, Sigmund O (2011) https://doi.org/10.1007/s00158-010-0594-7	MATLAB (Octave compatible)	– 2D – Compliance minimization – Successor to top99 with speed-up and additional functionalities	– Sensitivity filter; density filter – Heaviside projection – Alternative implementations of filtering: Use "*conv2*" function or Helmholtz type PDE – SIMP interpolation – OC update scheme	– Linear elasticity – Direct solver – Quadrilateral discretization	– Vectorization of loops – Use of "*sparse*" function for assembly – Memory preallocation – Restructure the program by moving a maximum amount of code out of the optimization loop – Label: Vectorized	– Code available at www.topopt.dtu.dk – Ready to use – Built upon top99 – Provide guidelines to use alternative filter implementations – Provide continuation scheme in terms of Heaviside projection
Educational	"*PolyTop*" Talischi C, Paulino GH, Pereira A, Menezes IFM. (2012b) https://doi.org/10.1007/s00158-011-0696-x	MATLAB (Octave compatible)	– 2D – Compliance minimization – A general topology optimization framework using unstructured polygonal meshes	– Density filter – Heaviside projection – SIMP (or RAMP) interpolation – OC update scheme	– Linear elasticity – Direct solver – Polygonal discretization	– Use of "*sparse*" function for assembly – Label: Vectorized	– Code attached using ESM – Ready to use – Provide continuation scheme of penalty parameter – Decoupling of the update scheme from the analysis routine
Educational	"*top3d*" Liu K, Tovar A (2014) https://doi.org/10.1007/s00158-014-1107-x	MATLAB (Octave compatible)	– 3D – Compliance minimization; displacement maximization; heat conduction – An efficient and compact MATLAB code to solve 3D topology optimization problems	– Density filter; sensitivity filter; gray-scale filter – SIMP interpolation – SQP, MMA, or OC update scheme	– Linear elasticity – Direct solver; iterative solver – Hexahedral discretization	– Use of "*sparse*" function for assembly – Iterative solver using "*pcg*" function for large-scale problems – Label: Vectorized	– Code available at http://top3dapp.com – Ready to use – Built upon top88 – Provide continuation strategy of penalty parameter – Implementation of SQP or MMA available at http://top3dapp.com

Journal : Large 158 | Article No : 3050 | Pages : 54 | MS Code : 3050 | Dispatch : 28-9-2021

Table 1 (continued)

Type	Name/Reference	Environment	Summary	Techniques: Topology optimization	Techniques: FE method	Techniques: Efficiency	Usability
Educational	Lagaros ND, Vasileiou N, Kazakis G (2019) https://doi.org/10.1007/s11081-018-9384-7	SAP2000 (C#)	– 3D – Compliance minimization – A C# code interacted with SAP2000 based on open application programming interface to perform density-based topology optimization in 3D	– Density filter – SIMP interpolation – OC or MMA update scheme	– Linear elasticity – Direct solver – Hexahedral discretization	– Label: Loop-based	– Code available at https://github.com/nikoslagaros/TOCP – Need access to SAP2000 – Partially based on top99 and top88
Research	"*EGP*" Zeng Z, Ma F (2020) https://doi.org/10.1016/j.advengsoft.2020.102863	MATLAB (Octave incompatible: Invalid call to mean)	– 2D and 3D – Compliance minimization; displacement maximization – An efficient gradient projection (EGP) method	– Density filter using "*imgaussfilt*" (2D) and "*imgaussfilt3*" (3D) functions – SIMP interpolation – Gradient projection update scheme	– Linear elasticity – Direct solver – Quadrilateral and hexahedral discretization	– Gradient clipping strategy – Approximate the projection by an analytical expression – Simplify the calculation of searching steps – Label: Vectorized	– Code available at https://github.com/zengzhi2015/EGP – Ready to use – Built upon top88
Research	"*PolyTop3D*" Chi H, Pereira A, Menezes IFM, Paulino GH (2020) https://doi.org/10.1007/s00158-019-02268-w	MATLAB (Octave incompatible: Requires "*delaunayTriangulation*" function)	– 3D – Compliance minimization – VEM-based topology optimization framework on general polyhedral discretization	– Density filter – SIMP interpolation – OC update scheme	– Linear elasticity – Direct solver – Polyhedral discretization – Virtual element method (VEM)	– Use of "*sparse*" function for assembly – Use of virtual element method – Label: Vectorized	– Code attached using ESM – Ready to use – Modularized in a similar manner to the PolyTop code
Research	"*top2dmgcg*" and "*top3dmgcg*" Amir O, Aage N, Lazarov BS (2014) https://doi.org/10.1007/s00158-013-1015-5	MATLAB (Octave compatible)	– 2D and 3D – Compliance minimization; displacement maximization – Improve computational efficiency by exploiting the multigrid preconditioned conjugate gradients (MGCG) solver	– Density filter; sensitivity filter – SIMP interpolation – OC or MMA update scheme	– Linear elasticity – Iterative solver: MGCG solver – Quadrilateral and hexahedral discretization	– Use of the MGCG solver – The total number of MGCG iterations can be reduced by imposing a convergence criterion on the approximation of the design sensitivities – Label: Vectorized and optimized	– Codes attached using ESM and available at https://github.com/odedamir/topopt-mgcg-matlab – Ready to use – Built upon top88

Journal : Large 158 | Article No : 3050 | Pages : 54 | MS Code : 3050 | Dispatch : 28-9-2021

Table 1 (continued)

Type	Name/Reference	Environment	Summary	Techniques			Usability
				Topology optimization	FE method	Efficiency	
Research	"*MinV*" Amir O (2015) https://doi.org/10.1007/s00158-014-1098-7	MATLAB (Octave compatible)	– 2D and 3D – Volume minimization – Efficient optimization procedure with recycled preconditioning	– Density filter – SIMP interpolation – OC update scheme	– Linear elasticity – Reanalysis-based approach (2D) and recycled preconditioning within a MGCG solver (3D) – Quadrilateral and hexahedral discretization	– The volume minimization formulation with recycled preconditioning is more efficient than a general compliance minimization formulation – Use of the MGCG solver – Label: Vectorized and optimized	– Code available at https://structopt.net.technion.ac.il/software/matlab-codes/ – Ready to use
Educational	"*top99neo*" and "*top3D125*" Ferrari F, Sigmund O (2020) https://doi.org/10.1007/s00158-020-02629-w	MATLAB (Octave incompatible: "*fsparse*" is incompatible)	– 2D and 3D – Compliance minimization – An exemplary code collecting the newest shortcuts and speed-ups to tackle medium-/large-scale topology optimization problems efficiently and additional functionalities	– Density filter using "*imfilter*" function – Heaviside projection – SIMP interpolation – OC update scheme with accelerations	– Linear elasticity – Equation solver using "*decomposition*" for 2D and "*chol*" for 3D – Quadrilateral and hexahedral discretization	*Speed-up matrix assembly:* – Define mesh-related quantities as integers(*int32*) – Use an assembly routine "*fsparse*" – Assemble one half of the matrix *Speed-up of the OC update:* – Use volume-preserving filtering schemes – Estimate the interval bracketing the current Lagrange multiplier – Explicit expression of the Lagrange multiplier *Acceleration of the OC iteration:* – Use PAE (periodic Anderson extrapolation) to accelerate convergence – Label: Vectorized and optimized	– Code available at www.topopt.dtu.dk – Require "*stenglib*" package to use "*fsparse*" function (available at https://github.com/stefanengblom/stenglib)

Journal : Large 158 | Article No : 3050 | Pages : 54 | MS Code : 3050 | Dispatch : 28-9-2021

Table 1 (continued)

Type	Name/Reference	Environment	Summary	Techniques			Usability
				Topology optimization	FE method	Efficiency	
Educational	"*TopOpt_in_PETSc*" Aage N, Andreassen E, Lazarov BS (2015) https://doi.org/10.1007/s00158-014-1157-0	PETSc (C++)	– 3D – Compliance minimization; homogenization problems (isotropic Poisson's ratio minimization and bulk modulus maximization) – An easy-to-use, fully parallelized and open-source framework for large-scale topology optimization	– Density filter; sensitivity filter; PDE filter – SIMP interpolation – MMA update scheme	– Linear elasticity (compliance minimization); elastic material design (homogenization problems) – Linear solver: Galerkin projection multigrid preconditioned flexible GMRES with GMRES/SOR smoothing – Hexahedral discretization	– The use of parallel and scientific computing – Label: Parallelized	– Code available at www.topopt.dtu.dk/PETSc – Need access to PETSc – The implementation is parallel scalable to thousands of cores and portable to Linux, UNIX, Mac and Windows
Educational	"*TopADD*" Zhang ZD, Ibhadode O, Bonakdar A, Toyserkani E (2021) https://doi.org/10.1007/s00158-021-02917-z	PETSc (C++)	– 2D and 3D – Compliance minimization; displacement maximization; heat conduction – A parallel-computing framework for 2D and 3D topology optimization with arbitrary design domains	– Density filter; sensitivity filter; PDE filter – Heaviside projection – SIMP interpolation – MMA update scheme	– Linear elasticity – Linear solver: Galerkin projection multigrid preconditioned flexible GMRES with GMRES/SOR smoothing – An efficient voxelizer to initialize arbitrary geometry as the design domain – Quadrilateral and hexahedral discretization	– The use of parallel and scientific computing – Label: Parallelized	– Code available at https://github.com/wonderfulzzd/TopADD_2D_3D_Arbitrary_TopOpt_in_PETSc – Need access to PETSc – Built upon TopOpt_in_PETSc
Research	Schmidt S, Schulz V (2011) https://doi.org/10.1007/s00791-012-0180-1	CUDA (C++)	– 3D – Compliance minimization – Topology optimization on CUDA-enabled graphics cards	– Sensitivity filter – SIMP interpolation – OC update scheme	– Linear elasticity – Iterative solver: matrix-free conjugate gradient method – Hexahedral discretization	– The GPU code is found to be extremely efficient, being faster than a 48-core shared memory CPU system – Label: Parallelized	– Code available at http://www.mathematik.uni-trier.de/~schmidt/gputop

Journal : Large 158 | Article No : 3050 | Pages : 54 | MS Code : 3050 | Dispatch : 28-9-2021

Table 1 (continued)

Type	Name/Reference	Environment	Summary	Techniques			Usability
				Topology optimization	FE method	Efficiency	
Research	"*TopologyGAN*" Nie Z, Lin T, Jiang H, Kara LB (2021) https://doi.org/10.1115/1.4049533	TensorFlow (Python)	– 2D – Compliance minimization – A deep learning-based generative model for topology optimization (TopologyGAN)	– A design of the input matrices involving the initial physical fields – A hybrid generator architecture: U-SE-ResNet – Ground truth data generated by the SIMP method	– Linear elasticity – Quadrilateral discretization	– Label: Machine learning-based	– Code available at https://github.com/zhenguonie/2020_TopologyGAN – Need access to TensorFlow
Research	"*TOuNN*" Chandrasekhar A, Suresh K (2021) https:// doi.org/10.1007/s00158-020-02629-w	Python and C++	– 2D and 3D – Compliance minimization – Topology optimization using neural networks (TOuNN)	– Neural networks (NN) activation functions to represent the density field – Loss function defined based on the penalty formulation – Implicit filtering – Built-in backpropagation for sensitivity analysis – Optimization using NN	– Linear elasticity – Direct solver (Cholesky factorization) and assembly free deflated FE solver – Quadrilateral and hexahedral discretization	– Label: Machine learning-based	– Code (2D) available at www.ersl.wisc.edu/software/TOuNN.zip – Need access to pyTorch and CVXOPT libraries

Journal : Large 158 | Article No : 3050 | Pages : 54 | MS Code : 3050 | Dispatch : 28-9-2021

it is found that the GPU implementation has higher efficiency compared with the CPU implementation on a 48-core shared memory system. Finally, we review two contributions employing machine learning, which is an emerging new direction in the SMO community. Note that the main focus of the two papers is not to develop standard SIMP codes. Nie et al. (2021) developed a deep learning-based generative model (`TopologyGAN`) in Python to accelerate the topology optimization, where the ground truth data are generated using the SIMP method. Chandrasekhar and Suresh (2021) developed a framework (i.e., `TOuNN`) in Python and C++ to implement topology optimization directly using neural networks. The formulation is developed based on the SIMP method with the use of neural networks activation functions to represent the density field.

From the perspective of user experience, the standard SIMP codes reviewed above are established with well-organized structures and the educational ones are accompanied by sufficient explanations. Most of them are ready-to-use while a few of them require prior setup to make use of advanced libraries. In Table 1, the efficiency of the codes are labeled as "loop-based", "vectorized", "vectorized and optimized", "parallelized", and "machine learning-based". The label "loop-based" refers to codes employing "`for`" loops for the matrix assembly or objective and sensitivity computations (e.g., `top99`). The label "vectorized" denotes that the loops are vectorized to improve the efficiency, and the matrices are assembled based on a triplet form and via "`sparse`" function in MATLAB. The label "vectorized and optimized" refers to codes with further optimized speed-up techniques. The label "parallelized" indicates codes incorporating parallel computation techniques, and the label "machine learning-based" refers to codes leveraging machine learning techniques. We observe that, in general, the efficiency is improved in the above order of at least the first four labels (versatility and reliability of learning-based approaches still remain to be proven), albeit paying increasing cost on readability in the same order. We recommend the newcomers to start from the standard tutorial codes that focus on educating the method. In addition, we provide an illustrative figure (Fig. 4) demonstrating the evolution of the SIMP codes built upon standard codes to assist the learning process.

2.1.2 Density-based (SIMP) codes targeting specialized problems

Based on the standard educational codes mentioned in the preceding subsection, many studies developed educational/research codes to solve specialized design problems, such as considering multiple scales, multiple physics, multiple materials, reliability, buckling criteria, stress constraint (or objective), material/geometric nonlinearity, local geometric control, and structural dynamics. Tables 2, 3 and 4 present these codes and related techniques summarized from 13 educational, 12 research, and 1 review papers. Because those specialized codes aim at solving various complex problems and are not intended for maximizing computational efficiency, the efficiency-related techniques are not summarized in the tables.

The codes targeting multi-scale and multiphysics topology optimization problems are summarized in Table 2. Topology optimization of multi-scale problems, which typically refers to maximizing or minimizing macroscopic properties by topologically optimizing either micro-structures or both macro- and micro-structures concurrently, has been of growing interest. Xia and Breitkopf (2015) developed an educational MATLAB code based on `top88` to generate micro-structures in 2D, aiming at achieving extreme material properties using numerical homogenization methods. Additionally, Gao et al. (2019b) developed educational MATLAB codes (in both 2D and 3D), built upon `top88` and `top3d`, respectively, to concurrently optimize both micro- and macro-structures based on numerical homogenization approaches. Wu et al. (2021) provided a homogenization-based topology optimization code (built upon `top88`) in which 2D structures with minimized compliance are designed using optimal rank-2 micro-structures. We remark that although the provided code employs a homogenization-based topology optimization approach, it has many similarities with the SIMP method (e.g., element-wise interpolation of material properties) and tackles multi-scale design problems. Thus, we include the paper in this section.

Multiphysics topology optimization, tackling design problems with coupled or uncoupled physics fields (other than solely solid mechanics), has attracted increasing attention with a considerable number of open-source codes published in the literature. Regarding piezoelectric design problems, Homayouni-Amlashi et al. (2021) developed 2D topology optimization educational MATLAB codes to design piezoelectric actuators and energy harvesters based on `top88` with guidelines of tuning penalization parameters provided. For photonics design problems, Christiansen and Sigmund (2021a, 2021b) comprehensively developed both the theory and educational implementation tutorials for photonics inverse designs. A 200-line MATLAB code and five COMSOL Multiphysics models (COMSOL Multiphysics software, COMSOL AB 2021) are provided. Notably, through the comparison of using the gradient-based or nongradient-based optimizers, the authors illustrated the inappropriateness of applying nongradient-based approaches in large-scale topology optimization problems (Christiansen and Sigmund 2021b). In terms of fluid design problems involving Stokes flow, Olesen et al. (2006) performed topology optimization to minimize energy dissipation or maximize velocities at prescribed locations based on the software COMSOL Multiphysics (formerly FEMLAB). To enable a

Journal : Large 158 | Article No : 3050 | Pages : 54 | MS Code : 3050 | Dispatch : 28-9-2021

Table 2 Summary of density-based (SIMP) codes tackling multi-scale and multi-physics problems

Type	Name/References	Environment	Summary and Specialty	Techniques		Usability
				Topology optimization	FE method	
Educational (multi-scale)	"*topX*" Xia L, Breitkopf P (2015) https://doi.org/10.1007/s00158-015-1294-0	MATLAB	– 2D – Maximization or minimization of homogenized material properties – TO of micro-structures to design materials with extreme properties	– Sensitivity filter; density filter – SIMP interpolation – Energy-based homogenization method to evaluate effective material property – OC update scheme	– Linear elasticity – Direct solver – Quadrilateral discretization	– Code attached using images – Built upon top88
Educational (multi-scale)	"*ConTop2D*" and "*ConTop3D*" Gao J, Luo Z, Xia L, Gao L (2019) https://doi.org/10.1007/s00158-019-02323-6	MATLAB	– 2D and 3D – Compliance minimization – Concurrent TO design on micro- and macro-scales	– Sensitivity filter; density filter – Heaviside projection – SIMP interpolation – Energy-based homogenization method to evaluate effective property – OC update scheme	– Linear elasticity – Direct solver – Quadrilateral and hexahedral discretization	– Code attached using texts (users can copy-and-paste the texts to create code files) – Partial codes provided – Built upon top88 and top3D
Review (multi-scale)	"*topRank2*" Wu J, Sigmund O, Groen JP (2021) https://doi.org/10.1007/s00158-021-02881-8	MATLAB (Octave incompatible: Error message "*nonconformant arguments*")	– 2D – Compliance minimization – Homogenization-based TO to design macro-structures using optimal rank-2 micro-structures	– Sensitivity filter – Rank-2 material model to evaluate effective material property – OC update scheme	– Linear elasticity – Direct solver – Quadrilateral discretization	– Code attached using ESM – Ready to use – Built upon top88
Educational (multi-physics)	"*Piezo_Actuator*" and "*Piezo_EnergyHarvester*" Homayouni-Amlashi A, Schlinquer T, Mohand-Ousaid A, Rakotondrabe M (2021) https://doi.org/10.1007/s00158-020-02726-w	MATLAB (Octave compatible for the code "*Piezo_Actuator*")	– 2D – Displacement maximization; minimization of the weighted sum of mechanical and electrical energy – TO design of piezoelectric plate with actuation and energy harvesting	– Sensitivity filter; density filter – PEMAP-P (Piezoelectric material with penalization and polarization) interpolation – OC or MMA update scheme	– Linear elasticity – Direct solver – Quadrilateral discretization	– Code attached using ESM – Require "*mmasub.m*" and "*subsolv.m*" for the code "*Piezo_EnergyHarvester*" – Built upon top88 – Provide guidelines for tuning penalization parameters

Journal : Large 158 | Article No : 3050 | Pages : 54 | MS Code : 3050 | Dispatch : 28-9-2021

Table 2 (continued)

Type	Name/References	Environment	Summary and Specialty	Techniques: Topology optimization	Techniques: FE method	Usability
Educational (multi-physics)	"*top200EM*" Christiansen RE, Sigmund O (2021a, b) https://doi.org/10.1364/JOSAB.406048, https://doi.org/10.1364/JOSAB.405955	MATLAB; COMSOL (Octave incompatible: Requires "*optimoptions*" function)	– 2D – Maximization of the figure of merit (e.g., electromagnetic field intensity) – TO for the inverse design of nano-photonic structures	– Density filter – Heaviside projection – Different interpolation schemes depending on the problem at hand – Update schemes of gradient-based (e.g., "*fmincon*" in MATLAB) and non-gradient-based (e.g., "*ga*" in MATLAB) optimizers (for comparison purpose)	– Linear material – Direct solver – Quadrilateral discretization	– Code available at https://www.topopt.mek.dtu.dk – Ready to use (for the MATLAB code) – Need access to COMSOL (for the COMSOL code)
Research (multi-physics)	Olesen LH, Okkels F, Bruus H (2006) https://doi.org/10.1002/nme.1468	FEMLAB (Later known as COMSOL)	– 2D – Generic objective functions – TO design of fluid flow problem	– No filtering techniques – Interpolation for permeability coefficient – MMA update scheme	– Stokes flow – Direct solver – Quadrilateral or triangular discretization	– Code attached using texts (users can copy-and-paste the texts to create code files) – Require "*mmasub.m*" and "*subsolv.m*"
Research (multi-physics)	"*chebytop*" Evgrafov A (2015) https://doi.org/10.1007/s00158-014-1176-x	MATLAB (Octave incompatible: Requires "*ldl*" function)	– 2D – Minimization of dissipated energy – TO design of fluid flow problem with locally cubically convergence	– No filtering techniques – Interpolation for permeability coefficient – Update scheme based on Chebyshev's method	– Stokes flow – Direct solver – Quadrilateral discretization	– Code attached using ESM – Ready to use – Provide and test three strategies for tuning optimization parameters
Educational (multi-physics)	"*PolyTopFluid*" Pereira A, Talischi C, Paulino GH, Menezes IFM, Carvalho MS (2016) https://doi.org/10.1007/s00158-014-1182-z	MATLAB (Octave incompatible: Requires "*TriRep*" function)	– 2D – Minimization of the average pressure drop between the inlet and the outlet; maximization of velocity at prescribed regions – TO design of fluid flow problem	– No filtering techniques – Interpolation for permeability coefficient – Option to include design-dependent viscosity – OC update scheme	– Stokes flow – Direct solver – Polygonal discretization	– Code attached using images (also available at http://paulino.ce.gatech.edu/software.html) – Ready to use – Built upon PolyTop

Journal : Large 158 | Article No : 3050 | Pages : 54 | MS Code : 3050 | Dispatch : 28-9-2021

Table 2 (continued)

Type	Name/References	Environment	Summary and Specialty	Techniques		Usability
				Topology optimization	FE method	
Research (multi-physics)	Jensen KE (2018) https://doi.org/10.1016/j.compfluid.2018.07.011	MATLAB (Octave compatible)	– 2D and 3D – Minimization of viscous dissipation; minimization of the flow velocity at specific locations – Demonstrate that anisotropic mesh adaptation enables a better description of solid domains in TO of flow problems	– No filtering techniques – Interpolation for the Darcy number – Anisotropic mesh adaptation – OC update scheme; a new steepest descent optimizer for unconstraint reverse flow problem	– Stokes flow – Direct solver – Polygonal discretization	– Code available at https://github.com/KristianE86/trullekrul – Ready to use

Topology optimization is abbreviated as TO in this table

locally cubic convergence, Evgrafov (2015) proposed a new update scheme based on Chebyshev's method to minimize dissipated energy of Stokes flows. The MATLAB code is provided as ESM. Moreover, an educational MATLAB code `PolyTopFluid` (Pereira et al. 2016), built upon `PolyTop`, is developed to handle the optimization problems of minimizing power dissipation or maximizing velocities for Stokes flow at prescribed locations with FE analysis using polygonal finite elements. Jensen (2018) developed a MATLAB code for topology optimization of Stokes flow and demonstrated the advantages of using anisotropic mesh adaptation. All the four studies involving fluid flow topology optimization developed their codes upon the formulation proposed by Borrvall and Petersson (2003).

In addition, design problems considering multiple materials, reliability, structural buckling, and stress constraint (or objective) have also been tackled using the density-based SIMP method with open-source codes provided, as shown in Table 3. Efforts have been made to develop multi-material topology optimization, which enlarges the design space and is applicable to practical engineering problems. With a 115-line MATLAB code published as ESM (modified from `top88`), Tavakoli and Mohseni (2014) developed a multi-material topology optimization approach by solving a series of sub-problems with binary materials using the alternative active-phase algorithm. Tavakoli (2014) proposed a new computational algorithm based on a volume constrained Allen–Cahn system to solve multi-material problems with the MATLAB code built upon `top88`. Although this algorithm has similarities with the phase-field approach, it uses density-based SIMP interpolation. Thus, we report this paper in the density-based method section. Based on the multi-material formulation and the efficient ZPR (Zhang-Paulino-Ramos) update scheme proposed by Zhang et al. (2018), Sanders et al. (2018) developed an educational MATLAB code `PolyMat` built upon `PolyTop`, to solve multi-material design problems on 2D polygonal discretization with many volume constraints.

Regarding topology optimization problems considering uncertainties, Kharmanda et al. (2004) proposed a simplified reliability-based formulation by pre-computing reliability aspects before topology optimization and provided the MATLAB code (built upon `top99`) as ESM. Csébfalvi (2017) handled 2D and 3D compliance minimization problems with uncertain loading directions via robust topology optimization, in which the expected compliance function is derived analytically. MATLAB codes built upon `top88` and `top3D` are provided. Keshavarzzadeh et al. (2019) developed a topology optimization framework considering loading and geometric uncertainties with multi-resolution FE models to reduce computational cost. In terms of buckling-based topology optimization, Ferrari et al. (2021) provided a 250-line educational MATLAB code to handle topology

Journal : Large 158 | Article No : 3050 | Pages : 54 | MS Code : 3050 | Dispatch : 28-9-2021

Table 3 Summary of density-based (SIMP) codes tackling multi-material, reliability, buckling, and stress-based problems

Type	Name/References	Environment	Summary and specialty	Techniques		Usability
				Topology optimization	FE method	
Research (multi-material)	"*Alternating active-phase algorithm*" Tavakoli R,Mohseni SM (2014) https://doi.org/10.1007/s00158-013-0999-1	MATLAB (Octave incompatible: Requires "*imresize*" function)	– 2D – Structural compliance minimization; thermal compliance minimization – Multimaterial TO by solving a series of binary material optimization sub-problems	– Sensitivity filter – SIMP interpolation – Alternating active phase algorithm – OC update scheme	– Linear elasticity – Direct solver – Quadrilateral discretization	– Code attached using ESM – Ready to use – Built upon top88
Research (multi-material)	Tavakoli R (2014) https://doi.org/10.1016/j.cma.2014.04.005	MATLAB (Octave compatible)	– 2D – Compliance minimization – Multimaterial TO by volume constrained Allen–Cahn system and regularized projected steepest descent method	– The extended Modica and Mortola approach to avoid topological instability – SIMP interpolation – The fractional step projected steepest descent method (update scheme)	– Linear elasticity – Direct solver – Quadrilateral discretization	– Code attached using texts (users can copy-and-paste the texts to create code files) – Ready to use – Built upon top88
Educational (multi-material)	"*PolyMat*" Sanders ED, Pereira A, Aguilo MA, Paulino GH (2018) https://doi.org/10.1007/s00158-018-2094-0	MATLAB (Octave incompatible: Requires "*rng*" function)	– 2D – Compliance minimization – TO design with multiple materials and multiple volume constraints	– Density filter; sensitivity filter; ZPR (Zhang-Paulino-Ramos) filter – SIMP (or RAMP) and DMO (Discrete material optimization) multi-material interpolation – ZPR update scheme	– Linear elasticity – Direct FE solver – Polygonal discretization	– Code attached using ESM – Ready to use – Built upon PolyTop
Research (reliability)	"*Reliability-based topology optimization* " Kharmanda G,Olhoff N,Mohamed A,Lemaire M (2004) https://doi.org/10.1007/s00158-003-0322-7	MATLAB	– 2D – Compliance minimization – Integrate reliability constraint in deterministic TO	– Sensitivity filter – SIMP interpolation – Simplified reliability constraints – OC update scheme	– Linear elasticity – Direct solver – Quadrilateral discretization	– Code attached using texts (users can copy-and-paste the texts to create code files) – Encounter syntax error when using the attached code – Built upon top99
Research (reliability)	Csebfalvi A (2017) https://doi.org/10.3311/PPci.10214	MATLAB	– 2D and 3D – Expected compliance minimization – TO of structures cusing analytically determined exact objective functions	– Sensitivity filter – SIMP interpolation – Analytical approach to evaluate expected compliance – OC update scheme	– Linear elasticity – Direct solver – Quadrilateral and hexahedral discretization	– Code attached using texts (users can copy-and-paste the texts to create code files) – Partial codes provided – Built upon top88 and top3D

Journal : Large 158 | Article No : 3050 | Pages : 54 | MS Code : 3050 | Dispatch : 28-9-2021

Table 3 (continued)

Type	Name/References	Environment	Summary and specialty	Techniques		Usability
				Topology optimization	FE method	
Research (reliability)	"*TOPOPT-MR*" Keshavarzzadeh V,Kirby RM,Narayan A (2019) https://doi.org/10.1002/nme.6063	MATLAB	– 2D and 3D – Compliance minimization – TO considering uncertainty using multi-resolution FE models	– Density filter – Heaviside projection – SIMP interpolation – OC update scheme	– Linear elasticity – Bifidelity approximation – Direct solver – Quadrilateral and hexahedral discretization	– Code (2D) available at https://github.com/vahid28k/Parametric-TOPOPT-Multi-Resolution – Ready to use – Built upon top88
Educational (buckling)	"*topBuck250*" Ferrari F,Sigmund O,Guest JK (2021) https://doi.org/10.1007/s00158-021-02854-x	MATLAB	– 2D – Maximization of buckling load factors; volume minimization – TO design for problems with linearized buckling criteria	– Density filter – Heaviside projection – SIMP interpolation – Kreisselmeier-Steinhauser aggregation – OC or MMA update scheme	– Linear elasticity – Direct solver – Quadrilateral discretization	– Code attached using images – Built upon top99neo
Research (stress)	"*PTOs*" and "*PTOc*" Biyikli E,To AC (2015) https://doi.org/10.1371/journal.pone.0145041	MATLAB (Octave compatible)	– 2D – Compliance minimization; volume minimization with stress constraints – A novel TO algorithm: Proportional Topology Optimization (PTO)	– Density filter – SIMP interpolation – Proportional TO algorithm (update scheme)	– Linear elasticity – Direct solver – Quadrilateral discretization	– Code available at www.ptomethod.org – Ready to use – Built upon top88
Educational (stress)	"*PolyStress*" Giraldo-Londono O,Paulino GH (2021b) https://doi.org/10.1007/s00158-020-02760-8	MATLAB (Octave compatible)	– 2D – Mass minimization – TO with local stress constraints handled by the augmented Lagrangian method	– Density filter – SIMP interpolation – Augmented Lagrangian method – MMA update scheme	– Nonlinear elasticity – Iterative solver – Polygonal discretization	– Code attached using ESM – Ready to use – Built upon PolyTop
Educational (stress)	Deng H,Vulimiri PS,To AC (2021) arXiv:210401210	MATLAB	– 2D and 3D – Stress minimization – Derivation and explanation for sensitivity analysis of stress-based TO with educational purposes	– Density filter – SIMP interpolation – P-norm stress aggregation – MMA update scheme	– Linear elasticity – Direct solver – Quadrilateral and hexahedral discretization	– Code not downloadable in June,2021

Topology optimization is abbreviated as TO in this table

Journal : Large 158 | Article No : 3050 | Pages : 54 | MS Code : 3050 | Dispatch : 28-9-2021

Table 4 Summary of density-based (SIMP) codes tackling local geometric control, material and geometric nonlinearities, and dynamic problems

Type	Name/Reference	Environment	Summary and specialty	Techniques: Topology optimization	Techniques: FE method	Usability
Research (local geometric control)	Fernandez E, Collet M, Alarcon P, Bauduin S, Duysinx P (2019) https://doi.org/10.1007/s00158-019-02313-8	MATLAB	– 2D and 3D – Structural compliance minimization; thermal compliance minimization; displacement maximization – Investigate aggregation strategy for maximum size local constraints	– Density filter – Heaviside projection – SIMP interpolation – Local maximum size constraint with specified test regions – P-mean or p-norm aggregation strategies – MMA update scheme	– Linear elasticity – Direct solver – Quadrilateral discretization	– Code (2D) attached using texts (users can copy-and-paste the texts to create code files) – Partial codes provided – Built upon PolyTop
Educational (nonlinearity)	"*TOGN213*" Chen Q, Zhang X, Zhu B (2019) https://doi.org/10.1007/s00158-018-2138-5	MATLAB interacted with ANSYS parametric design language	– 2D – Compliance minimization; displacement maximization – TO with hyperelastic materials and large deformations	– Density filter – SIMP interpolation – Additive hyperelasticity technique to circumvent numerical difficulties under large deformations – MMA update scheme	– Nonlinear elasticity with large deformations – Iterative solver – Quadrilateral discretization	– Code attached using images – Need access to ANSYS
Research (nonlinearity)	Dunning PD (2020) https://doi.org/10.1007/s00158-020-02605-4	MATLAB	– 2D – Compliance minimization; complimentary work minimization; displacement maximization – Investigate the co-rotational method to solve geometrically nonlinear TO problems	– Density filter – SIMP interpolation – MMA update scheme	– Linear elasticity with large deformations – Iterative solver – Co-rotational method to construct the tangent matrix – Quadrilateral discretization and 3-node shell element	– Code attached using images – Partial codes provided – Built upon top88
Educational (nonlinearity)	Zhu B, Zhang X, Li H, Liang J, Wang R, Li H, Nishiwaki S (2021) https://doi.org/10.1007/s00158-020-02733-x	FreeFEM++	– 2D – Compliance minimization – TO with large deformations	– Density filter – Heaviside projection – SIMP interpolation – Stored-energy interpolation to circumvent numerical difficulties under large deformations – OC update scheme	– Linear elasticity with large deformations – Iterative solver – Triangular discretization	– Code attached using images – Provide a strategy to overcome the nonconvergence problem
Research (dynamics)	"*SMC*" Martin A, Deierlein GG (2020) https://doi.org/10.1016/j.engstruct.2020.110717	MATLAB	– 2D and 3D – Sum of modal compliances (SMC) minimization – Dynamic topology optimization using modal decomposition	– Density filter – SIMP interpolation – OC update scheme	– Linear elasticity – Modal response spectrum analysis – Direct solver – Polygonal discretization	– Code (2D) attached using ESM – Partial codes provided – Built upon PolyTop

Journal : Large 158 | Article No : 3050 | Pages : 54 | MS Code : 3050 | Dispatch : 28-9-2021

Table 4 (continued)

Type	Name/Reference	Environment	Summary and specialty	Techniques: Topology optimization	Techniques: FE method	Usability
Educational (dynamics)	"*PolyDyna*" Giraldo-Londono O, Paulino GH (2021a) https://doi.org/10.1007/s00158-021-02859-6	MATLAB (Octave compatible)	– 2D – Dynamic compliance minimization; optimization of mean strain energy; minimization of mean squared displacement – TO of structures subjected to dynamic loads	– Density filter – Heaviside projection – SIMP (or RAMP) interpolation – "Discretize-then-differentiate" approach to evaluate the sensitivity – ZPR update scheme	– Linear elasticity – Direct solver – HHT-α method – Polygonal discretization	– Code attached using ESM – Ready to use – Built upon PolyTop

Topology optimization is abbreviated as TO in this table

optimization problems with linearized buckling criteria and included stiffness, volume, and buckling load factors either as the objective function or as constraints. This code makes use of the speed-up techniques in `top99neo` code to enable high-efficiency computation. In the area of stress-based topology optimization, Biyikli and To (2015) developed the proportional topology optimization (PTO) method for stress constrained and minimum compliance problems, where the design variables are assigned to elements proportionally to the value of stress or compliance. Two individual MATLAB codes, `PTOs` and `PTOc`, are provided, respectively. These codes are built upon `top88` with the OC algorithm replaced by the PTO and other modifications to add stress analysis and remove sensitivity analysis. Nevertheless, it should be noted that although the proposed PTO does not employ formal sensitivity analysis, it uses stress in the optimization update, which is analogous to using gradients for the compliance problem. Moreover, the use of a fully stressed design strategy does not result in stress optimal designs, as shown in Zhou and Sigmund (2017). Giraldo-Londoño and Paulino (2021b) developed an educational MATLAB code (`PolyStress`) built upon `PolyTop` for topology optimization with many local stress constraints handled by the augmented Lagrangian method. The `PolyStress` considers both linear and nonlinear material properties and provides a library of benchmark problems. Deng et al. (2021) developed a 146-line educational MATLAB code for 3D stress-minimization topology optimization and thoroughly discussed the sensitivity analysis in the paper.

Topology optimization with local geometric control, material and geometric nonlinearities, and structural dynamics using the density-based SIMP method are summarized in Table 4. Integrating geometric controls into topology optimization allows for designs possessing desired geometric features. To control maximum member sizes for the optimized designs, Fernández et al. (2019) adopted local geometric constraints that are formulated into a single constraint through different aggregation functions (i.e., p-norm and p-mean functions). A MATLAB code developed upon `PolyTop` is provided. Another practical design consideration in topology optimization is material and geometric nonlinearity. Employing the FE analysis module in ANSYS (Ansys Inc. 2021) through APDL (ANSYS parametric design language), Chen et al. (2019) developed a 213-line educational MATLAB code for topology optimization of hyperelastic materials under large deformations. Dunning (2020) adopted a co-rotational method, enabling the tangent stiffness matrix to be positive definite, and performed topology optimization under large deformations. The authors provided partial MATLAB codes (that can be used to modify `top88`) for implementations. Zhu et al. (2021) developed an 89-line educational MATLAB code for geometrically nonlinear structural topology optimization implemented in

Journal : Large 158 | Article No : 3050 | Pages : 54 | MS Code : 3050 | Dispatch : 28-9-2021

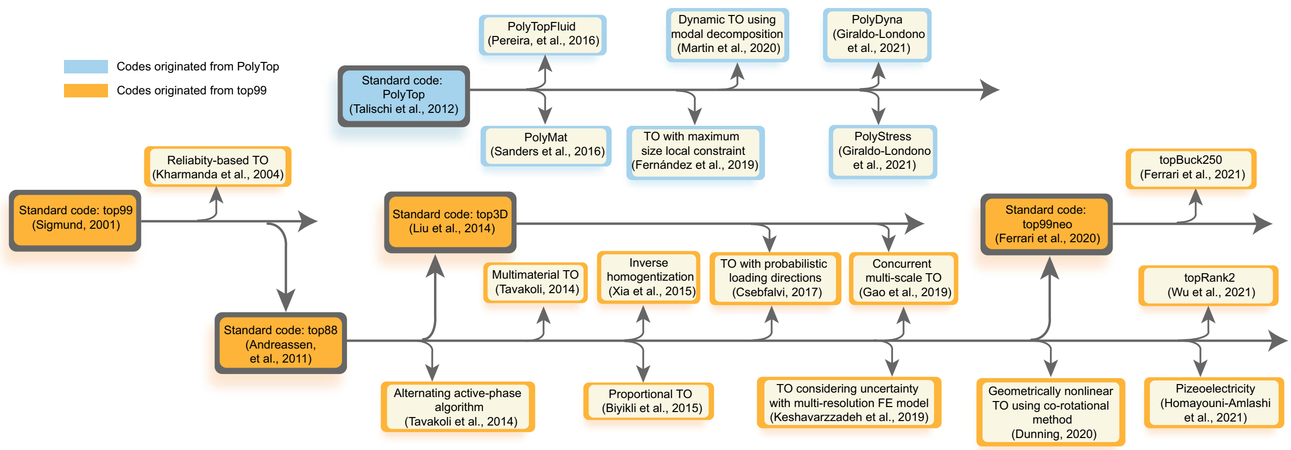


Fig. 4 Evolution of the specialized (SIMP) codes built upon standard codes. (Topology optimization is abbreviated as TO in this figure.)

FreeFEM++ (Hecht 2012) (which is an open-source program platform developed for numerically solving partial differential equations). For dynamic topology optimization problems, Martin and Deierlein (2020) proposed a sum dynamic compliance (SDC) method based on modal decomposition. The implementation of the proposed method is developed based on `PolyTop`, and partial MATLAB code (realizing the modal response spectrum analysis) is provided as ESM. Giraldo-Londoño and Paulino (2021a) developed an educational MATLAB code, built upon `PolyTop`, for dynamic topology optimization using HHT-α method. The code is named `PolyDyna` and provided using ESM.

We close this subsection by summarizing several user experiences on the density-based SIMP codes for specialized problems: (1) Many specialized SIMP codes are built upon the standard SIMP codes reviewed in Sect. 2.1.1, leading to a smooth learning curve for users, particularly for those who have prior experiences with standard SIMP codes. We illustrate this observation in Fig. 4. (2) To avoid confusion, when possible, educational codes are recommended to make use of consistent sensitivity analysis instead of shortcuts, such as fully stressed design rules or neglecting adjoint terms in, e.g., buckling problems. (3) Some specialized SIMP codes are provided using nontext format (e.g., image), which is not directly usable (i.e., requiring users to manually retype the codes). (4) Some specialized SIMP codes written in MATLAB are not compatible with Octave (the compatibility is reported in Tables 2, 3, 4).

2.1.3 Density-based discrete variable codes

Different from the SIMP material interpolation scheme in which the design variables are continuous, discrete variable topology optimization directly tackles the 0-1 design problem with material density being either void or solid. Papers on discrete variable approaches that provide open-source codes can be categorized into two classes. The first class constructs the optimization formulation based on integer programming, while the second one drives the optimization process based on an "evolutionary" metaphor, hence the name ESO (evolutionary structural optimization). Developed upon the original ESO approach that only removes inefficient material, its bi-directional version (BESO) can evolve the designs by adding and removing material simultaneously. The summary of 6 educational, 5 research, 1 review, and 1 forum discussion papers belonging to these two categories is shown in Tables 5 and 6.

In terms of the discrete topology optimization using integer programming, Liang and Cheng (2020) developed a 128-line educational MATLAB code that approximates the 0–1 design problem by a sequence of discrete variable sub-programming problems and solves these sub-programming problems by a Canonical relaxation algorithm. In the proposed formulation, a move limit strategy is employed to achieve a gradual volume reduction, which is derived from the similar essential technique for all ESO/BESO methods. Different from the ESO/BESO methods, the strategy in this paper requires the results to be converged in the intermediate volume fractions. Picelli et al. (2021) presented a 101-line educational MATLAB code with the implementation of the topology optimization of binary structures (TOBS) method composed of sequential linearization, constraints' relaxation, sensitivity filtering, and an integer programming solver. In this code, the integer programming sub-problems are solved via the branch-and-bound algorithm implemented in the MATLAB built-in function `intlinprog`. An alternative optimizer CPLEX©, which is a proprietary optimization package from IBM, is also recommended for a more efficient and robust branch-and-bound implementation. Souza et al. (2021) extended the TOBS method to

Table 5 Summary of density-based discrete variable topology optimization codes using integer programming

Type	Name/Reference	Environment	Summary	Techniques		Usability
				Topology optimization	FE method	
Educational	"*DVTOPCRA*" Liang Y, Cheng G (2020) https://doi.org/10.1007/s00158-019-02396-3	MATLAB (Octave compatible)	– 2D – Compliance minimization – Investigate discrete variable topology optimization with sequential integer programming and Canonical relaxation algorithm	– Sensitivity filter – SIMP interpolation (used in sensitivity analysis) – Update scheme based on Canonical relaxation algorithm and sequential integer programming	– Linear elasticity – Direct solver – Quadrilateral discretization	– Code attached using ESM – Ready to use
Educational	"*tobs101*" Picelli R, Sivapuram R, Xie YM (2021) https://doi.org/10.1007/s00158-020-02719-9	MATLAB (Octave incompatible: Requires "*optimoptions*" function)	– 2D – Compliance minimization – Propose topology optimization of binary structures (TOBS) method	– Sensitivity filter – SIMP interpolation (used in sensitivity analysis) – Update scheme based on integer linear programming	– Linear elasticity – Direct solver – Quadrilateral discretization	– Code attached using ESM – Ready to use – Built upon top88
Research	Souza B, Yamabe P, Sa L, Ranjbarzadeh S, Picelli R, Silva E (2021) https://doi.org/10.1007/s00158-021-02910-6	Python; Octave	– 2D and 3D – Minimization of dissipated energy; diodicity optimization problem – Extend the Topology Optimization of Binary Structures (TOBS) for fluid flow design	– No filtering techniques – Interpolation for permeability coefficient – Integer linear programming (CPLEX©optimization package)	– Stokes flow – Implemented in FEniCS – Triangular and tetrahedral discretization	– Code available at http://github.com/bruno-caldas/tobs – Need access to FEniCS – Provide guidance for tuning optimization parameters

Journal : Large 158 | Article No : 3050 | Pages : 54 | MS Code : 3050 | Dispatch : 28-9-2021

Table 6 Summary of density-based discrete variable topology optimization codes using the ESO/BESO methods

Type	Name/Reference	Environment	Summary	Techniques: Topology optimization	Techniques: FE method	Usability
Forum discussion	"*soft-kill BESO*" Huang X, Xie YM (2010) https://doi.org/10.1007/s00158-010-0487-9	MATLAB	– 2D – Compliance minimization – Response critical comments of BESO by comparing "soft-kill" BESO and the SIMP method	– Sensitivity filter – Average historical sensitivities – BESO update scheme	– Linear elasticity – Direct solver – Quadrilateral discretization	– Code attached using images – Built upon top99
Research	Zhou S, Cadman J, Chen Y, Li W, Xie YM, Huang X, Appleyard R, Sun G, Li Q (2012) https://doi.org/10.1016/j.ijheatmasstransfer.2012.08.028	MATLAB	– 2D and 3D – Minimization of difference between effective and target transport properties – BESO to design material unit cells with target transport properties (e.g., conductivity)	– Sensitivity filter – Average historical sensitivities – BESO update scheme – Homogenization method based on asymptotic analysis to evaluate effective properties	– Linear material – Direct solver – Quadrilateral and hexahedral discretization	– Code (2D) attached using images
Educational	Zuo ZH, Xie YM (2015) https://doi.org/10.1016/j.advengsoft.2015.02.006	Python	– 3D – Compliance minimization – BESO code communicating with ABAQUS (as a FE solver)	– Sensitivity filter – Average historical sensitivities – BESO update scheme	– Linear elasticity – Direct solver – Hexahedral discretization	– Code available at https://www.isg.rmit.edu.au – Need access to ABAQUS
Educational	"*sera*" Loyola RA, Querin OM, Jimenez AG, Gordoa CA (2018) https://doi.org/10.1007/s00158-018-1939-x	MATLAB (Octave compatible)	– 2D – Compliance minimization; displacement maximization – Sequential element rejection and admission (SERA) method developed upon BESO	– Sensitivity filter – Define "virtual material" to avoid using intermediate densities – SERA algorithm (update scheme)	– Linear elasticity – Direct solver – Quadrilateral discretization	– Code attached using texts (users can copy-and-paste the texts to create code files) – Ready to use – Progression, smoothing and material redistribution ratios may need adjustment for stable convergence
Review	"*esoL*" and "*esoX*" Xia L, Xia Q, Huang X, Xie YM (2018) https://doi.org/10.1007/s11831-016-9203-2	MATLAB	– 2D – Compliance minimization; maximization or minimization of homogenized material properties – BESO for benchmark designs of structures and material microstructures	– Sensitivity filter – Average historical sensitivities – BESO update scheme	– Linear elasticity – Direct solver – Quadrilateral discretization	– Code attached using images – Built upon top88

Journal : Large 158 | Article No : 3050 | Pages : 54 | MS Code : 3050 | Dispatch : 28-9-2021

Table 6 (continued)

Type	Name/Reference	Environment	Summary	Techniques		Usability
				Topology optimization	FE method	
Research	"*ETO*" Da D, Xia L, Li G, Huang X (2018) https://doi.org/10.1007/s00158-017-1846-6	MATLAB	– 2D and 3D – Compliance minimization; natural frequency maximization – Propose the evolutionary topology optimization (ETO) method to optimize structures with smoothed boundary representation using level-set functions	– Sensitivity filter – Average historical sensitivities – Level set function constructed based on the nodal sensitivity numbers – ETO update scheme	– Linear elasticity – Direct solver – Quadrilateral discretization	– Code (2D) attached using images – Built upon top88
Educational	"*DER-BESO*" Lin H, Xu A, Misra A, Zhao R (2020) https://doi.org/10.1007/s00158-020-02588-2	ANSYS parametric design language	– 2D and 3D – Compliance minimization – BESO with dynamic evaluation rate strategy	– Sensitivity filter – Average historical sensitivities – BESO update scheme with dynamic evolution rate strategy	– Linear elasticity – Direct solver – Quadrilateral and hexahedral discretization	– Code (2D) attached using images – Need access to ANSYS
Research	"*SEMDOT*" Fu YF, Rolfe B, Chiu LNS, Wang Y, Huang X, Ghabraie K (2020) https://doi.org/10.1016/j.advengsoft.2020.102921	MATLAB	– 2D – Compliance minimization – Elemental volume fractions based Smooth-Edged Material Distribution for Optimizing Topology (SEMDOT)	– Multiple filtering steps: Elemental volume fraction filter and heuristic filter – Heaviside smooth function – MMA update scheme	– Linear elasticity – Direct solver – Quadrilateral discretization	– Code attached using images – Require "*mmasub.m*" and "*subsolv.m*" – Built upon top88 and soft-kill BESO
Educational	"*TOP_Geo_Non*" Han Y, Xu B, Liu Y (2021b) https://doi.org/10.1007/s00158-020-02816-9	MATLAB	– 2D – Compliance minimization – BESO for geometrically nonlinear problems	– Sensitivity filter – Average historical sensitivities – BESO update scheme	– Linear elasticity under large deformations – Iterative solver – Quadrilateral discretization	– Code attached using images
Research	"*Soft_BESO_HFM*" Han H, Guo Y, Chen S, Liu Z (2021a) https://doi.org/10.1007/s00158-020-02771-5	MATLAB (Octave compatible)	– 2D – Compliance minimization – BESO with hole-filling method to achieve better control of the topological characteristics of optimized designs	– Sensitivity filter – Average historical sensitivities – Digital Gauss-Bonnet formula to count the number of holes in design domains – Hole-filling method to control the existence of holes – BESO update scheme	– Linear elasticity – Direct solver – Quadrilateral discretization	– Code attached using ESM – Ready to use – Built upon Soft-BESO MATLAB code available at https://www.isg.rmit.edu.au

Journal : Large 158 | Article No : 3050 | Pages : 54 | MS Code : 3050 | Dispatch : 28-9-2021

handle fluid flow problems and provided a MATLAB code, in which FE analysis is performed using FEniCS (Langtangen and Logg 2017) and design variables are updated via the CPLEX©optimizer.

ESO/BESO approaches are based on the concept of gradual removal and/or addition of materials in the design domain, which naturally fall into the discrete variable category. Huang and Xie (2010) developed a "soft-kill" BESO method ("soft-kill" means that inefficient elements are not completely removed and remain as fictitious void elements) implemented in a MATLAB code and compared with the results generated through the SIMP method. Zhou et al. (2012) extended the "soft-kill" BESO method to design targeting effective transport properties (e.g., conductivity) and provided a MATLAB code handling both 2D and 3D problems. Zuo and Xie (2015) developed a 100-line educational Python code interfacing with ABAQUS (ABAQUS Inc. 2021) to topologically optimize 3D structures using the "soft-kill" BESO method. With the similar idea of removing and adding material in elements, Loyola et al. (2018) developed a sequential element rejection and admission (SERA) method, as an improved version of BESO by introducing the concept of "virtual material". A MATLAB code is provided for educational purposes in that paper. Xia et al. (2018) developed two MATLAB codes (named as `esoL` and `esoX`) to generate benchmark designs of structures and material micro-structures, respectively. The two codes are built upon `top88` and can be used as standard codes suitable for beginners to study the ESO/BESO method. To achieve smooth boundary representations of continuum structures, Da et al. (2018) adopted the level-set function to determine the structural topology with smooth boundary representation and proposed an evolutionary topology optimization (ETO) method based on the conventional BESO method. The corresponding MATLAB code, named as `ETO`, is built upon `top88` and provided in the Appendix. More recently, Lin et al. (2020) conducted BESO with dynamic evolution rate in both 2D and 3D and presented an implementation (named as `DER-BESO`) using ANSYS parametric design languages in which the FE analysis is conducted in ANSYS (Ansys Inc. 2021), which enhances the usability by avoiding sequential executions between the optimization program and the FE analysis software. Fu et al. (2020) proposed an elemental volume fraction-based optimization framework named smooth-edged material distribution for optimizing topology, where the elemental volume fractions are determined by densities at grid points. A MATLAB code (`SEMDOT`) based on the proposed framework is provided for educational purposes. We note that the framework is developed based on both the SIMP approach and the ETO method (using a BESO-based optimizer). To avoid a repeated review, we present the paper only in this section. Han et al. (2021b) developed a 137-line educational MATLAB code to handle geometrically nonlinear problems via the BESO method. To enable controllable topological characteristics of optimized designs generated from the BESO method, Han et al. (2021a) proposed a hole-filing method and topological constraints to limit the maximum number of holes in the optimized designs. A MATLAB code built upon `top88` is provided in the paper.

Our user experience with the codes using density-based discrete variable topology optimization is summarized as follows: (1) Some codes are provided as nontext format (e.g., image), which is not directly usable (requiring users to manually retype the codes). (2) Some MATLAB codes require additional functions to be compatible with Octave. (3) For ESO/BESO methods, we recommend beginners to learn and use the basic codes as a start.

2.2 Level-set and other differential equation-driven approaches

Educational work associated with topology optimization employing level-set and other differential equation-driven approaches is reviewed in this subsection, including 10 educational papers and 2 research papers.

In the level-set method, the structure is implicitly represented by a moving boundary of a scalar function (i.e., level-set function). The motion of boundaries is tracked with the notion of a velocity field that leads to change of boundary shape. The level-set function used to define the material, void, and interface is shown as follows:

$$\begin{cases} \Phi(\boldsymbol{x}) > 0; & \boldsymbol{x} \in \Omega \\ \Phi(\boldsymbol{x}) = 0; & \boldsymbol{x} \in \partial\Omega \\ \Phi(\boldsymbol{x}) < 0; & \boldsymbol{x} \in D \setminus \Omega, \end{cases} \tag{4}$$

where $\Phi(\boldsymbol{x})$ is the level-set function, D is the design domain, Ω is the material domain, $D \setminus \Omega$ is the void domain, and $\partial\Omega$ is the material interface.

Traditionally, in the level-set-based topology optimization, the level-set function is updated via the solution of the Hamilton–Jacobi equation (Osher and Sethian 1988; Osher and Fedkiw 2006; Wang et al. 2003), which is formulated by taking derivative on both sides of the boundary equation as (Sigmund and Maute 2013),

$$\frac{\partial \Phi}{\partial t} + V|\nabla \Phi| = 0, \tag{5}$$

where t is a pseudo time representing the evolution during the optimization process, V is the velocity function determining the motion of the geometric boundary, which

Journal : Large 158 | Article No : 3050 | Pages : 54 | MS Code : 3050 | Dispatch : 28-9-2021

is dependent on the shape derivative of the optimization objective. To improve the numerical stability, regularize the level-set function, and to introduce holes inside the domain, the general Hamilton–Jacobi equation can be augmented by diffusive and reaction terms as follows (Sigmund and Maute 2013),

$$\frac{\partial \Phi}{\partial t} + V|\nabla \Phi| - \mathscr{D}(\Phi) - \mathscr{R}(\Phi) = 0. \tag{6}$$

The above equation is known as the generalized Hamilton–Jacobi equation, where $\mathscr{D}(\Phi)$ is the diffusive operator smoothing the level-set field, and $\mathscr{R}(\Phi)$ is the reactive term allowing for the nucleation of holes.

2.2.1 Level-set methods

In this subsection, we review the educational work dedicated to developing topology optimization codes using level-set methods, including 4 educational papers and 2 research papers (as shown in Table 7). Challis (2010) developed a discrete level-set-based topology optimization code (built upon the framework of density-based code `top99`). In this work, the reaction term driven by the topological derivative of the optimization objective is included to generate holes. An upwind finite difference scheme is utilized to solve the evolution equation. Laurain (2018) developed a FEniCS code for topology optimization based on the level-set method, where the Hamilton–Jacobi equation is solved via a forward Euler time discretization. Facilitated by the straightforward implementation of variational formulations using FEniCS, a distributed shape derivative is directly employed to compute a descent direction for the design objective. Chung et al. (2019) implemented both the density-based (SIMP method) and level-set-based topology optimization using OpenMDAO (Gray et al. 2019), which is an open-source multidisciplinary design optimization platform with a modular architecture. In the implementation of the level-set method, the boundary is updated based on the solution of the Hamilton–Jacobi equation. Taking advantage of the modularity, this framework can be easily reused or reconfigured, in which the reusability is exemplified by using pre-existing components to implement a new topology optimization formulation, and the reconfigurability is demonstrated by showing that the filtering can be easily changed via modifying only a few lines of codes. Alternative to directly solving the Hamilton–Jacobi equation, Wei et al. (2018) developed an 88-line MATLAB code using the parameterized level-set method (PLSM), where the level-set function is interpolated by a linear combination of radial basis functions (RBFs) and coefficients, and the parameterized coefficients at grid points are updated during the optimization process. The first-order forward Euler's method is employed to solve the evolution equation numerically. Liu et al. (2019) developed a subdomain parameterized level-set topology optimization framework using RBFs, where the global design domain is divided into a number of subdomains. In this way, the parameterization and evolution of level-set functions can be conducted in each subdomain separately. The effects of different RBF types, connectivity types of micro-structures, and subdomain size on the final optimized results were investigated. The 2D MATLAB implementation is provided as ESM. Andreasen et al. (2020) proposed a crisp interface level-set topology optimization approach using the cut element method. This study also illustrated the similarities and connections between the density field and the level-set field. Accordingly, the level-set formulation is established based on the density-based representation. In this case, the techniques of Heaviside projection (Wang et al. 2011) and MMA update scheme (Svanberg 1987) can be used for the level-set optimization approach, while the main differences lie in the modifications of FE and sensitivity analyses.

2.2.2 Other differential equation-driven approaches

In addition to the classical level-set approaches discussed in the preceding subsection, there are other differential equation-driven approaches (with a total of 6 educational papers) employing topological derivatives to generate the design. A brief summary is shown in Table 8.

The first four articles in Table 8 make use of the reaction-diffusion approach. In this approach, the convective term $V|\nabla \Phi|$ in the Hamilton–Jacobi equation (Eq. 6) is not included, and only the reaction term is employed to update the level-set function. In Yamada et al. (2010), the reaction term is formulated as a factor α multiplying the derivative of the objective functional, as shown below:

$$\frac{\partial \Phi(\boldsymbol{x})}{\partial t} = -\alpha \frac{\partial J}{\partial \Phi}, \tag{7}$$

where the objective functional J is defined as the summation of the elastic energy and the fictitious interfacial energy (Yamada et al. 2010), in which a regularization parameter representing the relative ratio between the fictitious interfacial energy and the elastic energy is introduced to realize a flexible control of the geometrical complexity of optimized structures. It is worth noting that whether this approach still belongs to the level-set method is debatable since it no longer employs the shape derivative information (Sigmund and Maute 2013). Following the work of Yamada et al. (2010), Otomori et al. (2015) developed a ready-to-use MATLAB code (i.e., `levelset88`) implementing

Table 7 Summary of level-set method topology optimization codes

Type	Name/Reference	Environment	Summary	Techniques: Topology optimization	Techniques: FE method	Usability
Educational	"*top_levelset*" Challis VJ (2010) https://doi.org/10.1007/s00158-009-0430-0	MATLAB (Octave compatible)	– 2D – Compliance minimization – A discrete level-set topology optimization code inspired by top99	– Discretization of the level-set function – Update level-set functions solving Hamilton–Jacobi equation with shape sensitivity and topological sensitivity – Reinitialization of level-set functions is required	– Linear elasticity – Direct solver – Quadrilateral discretization	– Code attached using ESM – Ready to use – Built upon top99 – Provide guidelines for tuning parameters – Can be used alongside top99 of the SIMP method to demonstrate similarities and differences between these two approaches
Educational	Laurain A (2018) https://doi.org/10.1007/s00158-018-1950-2	FEniCS (Python)	– 2D – Compliance minimization – A FEniCS code (written in Python) for structural optimization based on the level-set method	– Update level-set functions solving Hamilton–Jacobi equation – Use the distributed shape derivative to compute a descent direction for the compliance	– Linear elasticity – Triangular discretization	– Code attached using ESM – Need access to FEniCS – A step-by-step implementation is provided
Educational	Chung H, Hwang JT, Gray JS, Kim HA (2019) https://doi.org/10.1007/s00158-019-02209-7	OpenMDAO	– 2D – Compliance minimization – A modular paradigm for topology optimization using OpenMDAO	Density-based method: – Density filter – SIMP interpolation Level-set method: – Update level-set functions solving Hamilton–Jacobi equation – Sequential Quadratic Programming (SLSQP) from "*SciPy*" optimization library	– Linear elasticity – Quadrilateral discretization	– Code available at https://github.com/chungh6y/openmdao_TopOpt – Need access to OpenMDAO
Educational	"*TOPRBF*" Wei P, Li Z, Li X, Wang MY (2018) https://doi.org/10.1007/s00158-018-1904-8	MATLAB (Octave compatible)	– 2D – Compliance minimization – A code for the parameterized level-set method-based topology optimization using radial basis functions (RBFs)	– Parametrization: A level-set function is decoupled by a linear combination of a set of RBFs and coefficients – Evolution (updating) scheme: First-order forward Euler's method – Approximate re-initialization scheme	– Linear elasticity – Direct solver – Quadrilateral discretization	– Code attached using ESM – Ready to use – Built upon top88 – Provide guidelines for tuning parameters

Journal : Large 158 | Article No : 3050 | Pages : 54 | MS Code : 3050 | Dispatch : 28-9-2021

Table 7 (continued)

Type	Name/Reference	Environment	Summary	Techniques		Usability
				Topology optimization	FE method	
Research	"*Sub_LSM*" Liu H, Zong H, Tian Y, Ma Q, Wang MY (2019) https://doi.org/10.1007/s00158-019-02318-3	MATLAB (Octave compatible)	– 2D and 3D – Compliance minimization – A subdomain parameterized level-set topology optimization framework using radial basis functions (RBFs)	– Parameterized subdomain level set functions using RBFs – Update level-set functions in each subdomain separately and independently	– Linear elasticity – Direct solver – Quadrilateral discretization – Multi-node extended multi-scale FE method for the 3D layered cellular structures optimization	– Code (2D) attached using ESM – Ready to use – Built upon TOPRBF – Investigate effects of RBF types, connectivity types of microstructures, and subdomain size on the final optimized results
Research	"*CutTopOpt*" Andreasen CS, Elingaard MO, Aage N (2020) https://doi.org/10.1007/s00158-020-02527-1	MATLAB (Octave incompatible: '*GeometricConstraintInit*' undefined)	– 2D – Compliance minimization; displacement maximization – A crisp interface level-set optimization approach using a cut element method based on the ingredients from the density method	– Heaviside projection – MMA update scheme	– Linear elasticity – Direct solver – Quadrilateral discretization for background mesh; triangular discretization for cut elements	– Code available at www.topopt.dtu.dk. – Ready to use – Provide guidelines for tuning parameters

Journal : Large 158 | Article No : 3050 | Pages : 54 | MS Code : 3050 | Dispatch : 28-9-2021

Table 8 Summary of other differential equation-driven topology optimization codes

Type	Name/Reference	Environment	Summary	Techniques: Topology optimization	Techniques: FE method	Usability
Educational	"*levelset88*" Otomori M, Yamada T, Izui K, Nishiwaki (2015) https://doi.org/10.1007/s00158-014-1190-z	MATLAB	– 2D – Compliance minimization – Using a reaction-diffusion equation to update level-set functions	– Discretization of the level-set function – Update level-set functions using reaction-diffusion equation with topological sensitivity	– Linear elasticity – Direct solver – Quadrilateral discretization	– Code attached using images – Provide guidelines for tuning parameters – Built upon top88
Educational	"*filter_based_levelset*" Yaghmaei M, Ghoddosian A, Khatibi MM (2020) https://doi.org/10.1007/s00158-020-02540-4	MATLAB	– 2D – Total potential energy minimization; displacement maximization – A new level-set topology optimization method based on the filtration of the level-set function	– Filtered level-set function: "*imfilter*" – Level-set function penalization – Modified ALM update formula for Lagrange multiplier	– Linear elasticity – Direct solver – Quadrilateral discretization	– Code attached using images – Built upon levelset88
Educational	Liu Z, Korvink JG, Huang R (2005) https://doi.org/10.1007/s00158-004-0503-z	FEMLAB (Later known as COMSOL)	– 2D – Compliance minimization – A procedure implementing level-set-based topology optimization using the FEMLAB package	– Update level-set functions using reaction-diffusion equation	– Linear elasticity – Linear solver – Triangular discretization	– Code is not downloadable in July, 2021
Educational	Kim C, Jung M, Yamada T, Nishiwaki S, Yoo J (2020) https://doi.org/10.1007/s00158-020-02498-3	FreeFEM++	– 2D – Compliance minimization – An educational paper for reaction-diffusion equation based topology optimization	– Laplacian in the diffusion term (in update scheme) for filtering function – SIMP interpolation – Reaction-diffusion equation based update scheme	– Linear elasticity – Direct solver – Triangular discretization	– Code available at http://ssd.yonsei.ac.kr – Need access to FreeFEM++
Educational	"*UNVARTOP*" Yago D, Cante J, Lloberas-Valls O, Oliver J (2021) https://doi.org/10.1007/s00158-020-02722-0	MATLAB (Octave compatible)	– 2D – Compliance minimization; displacement maximization; multi-load compliance minimization – Implementation in MATLAB of the unsmooth variational topology optimization approach (UNVARTOP)	– Laplacian regularization filter – No material interpolation – Update scheme: cutting and bisection algorithm	– Linear elasticity – Direct solver – Quadrilateral discretization	– Code available at https://github.com/DanielYago/UNVARTOP – Ready to use
Educational	"*ParetoOptimalTracing*" Suresh K (2010) https://doi.org/10.1007/s00158-010-0534-6	MATLAB (Octave incompatible: '*contours*' undefined)	– 2D – Compliance-related multi-objective problem – A compact MATLAB code for generating Pareto-optimal topologies	– Topological sensitivity field filter – No material interpolation – Use Pareto-Frontier tracing algorithm to determine Pareto-optimal topologies	– Linear elasticity – Direct solver – Quadrilateral discretization	– Code available at www.mathworks.com/matlabcentral/fileexchange/ – Ready to use – Partially based on top99

Journal : Large 158 | Article No : 3050 | Pages : 54 | MS Code : 3050 | Dispatch : 28-9-2021

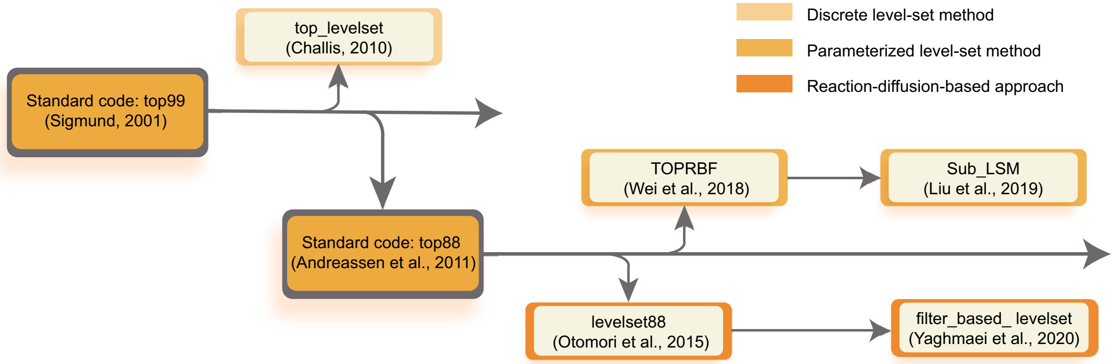


Fig. 5 Evolution of the level-set-based codes built upon standard codes

the topology optimization driven by the reaction-diffusion equation, in which the topological derivative of the compliance minimization problem is formulated. The effect of the regularization parameter on the geometrical complexity was examined via numerical examples. Instead of solving a set of linear equations to adjust the complexity of the configuration, Yaghmaei et al. (2020) developed a method using the filtration of the level-set function to control the optimized configuration, making the optimization process computationally efficient. The total potential energy minimization and compliant mechanism problems were investigated, and the corresponding topological derivatives were derived to measure the sensitivity of objective function with respect to the domain perturbation. Based on the proposed method, a compact MATLAB code derived from `top88` and `levelset88` is provided for educational purposes. Based on the reaction-diffusion approach, Liu et al. (2005) introduced an educational topology optimization procedure implementing the 2D compliance minimization problem using the FEMLAB (later known as COMSOL), where the reaction-diffusion equation is handled by a FE solver via the FEMLAB. However, this code is not reachable at present. Kim et al. (2020) implemented the reaction-diffusion equation-driven approach using FreeFEM++ (Hecht 2012), a free and user-friendly FE software, enabling high-resolution boundaries of the optimized structures using adaptive mesh refinement.

In terms of other educational studies using topological derivatives, Yago et al. (2021) developed a MATLAB code using the unsmooth variational topology optimization (UNVARTOP) method, where a relaxed topological derivative is formulated to serve as a directional derivative of the objective function. Based on the concept of topological sensitivity, Suresh (2010) developed an educational MATLAB code for the multi-objective topology optimization problems, where the Pareto-frontier tracing algorithm is utilized to determine Pareto-optimal topologies.

Based on the review of level-set and other differential equation-driven approaches (Tables 7 and 8), we summarize our observations and recommendations as follows. It is noted that in some studies the introduction of a set of tuning parameters related to different optimization algorithms may cause the results to be sensitive to the change of parameters (see Sect. 7). We recommend that educational papers to offer insights on the impact of these algorithmic tuning parameters on results, and provide instructions on parameter usage. In addition, for codes developed on platforms other than MATLAB/Octave, a step-by-step procedure should be included to ease the learning curve for users. Moreover, a downloadable and editable file format of the code is highly recommended to make it easily attainable for readers. To facilitate the learning process for beginners, we provide an illustrative figure (Fig. 5) demonstrating the evolution of several codes in the level-set-based approach.

2.3 Geometric component approaches

Several articles fall into the category of geometric component approaches. This subsection reviews codes in this category, which include 1 educational and 6 research papers.

The first one is the research work proposed by Saxena (2011) about topology optimization using negative masks overlay scheme based on gradient search. In the negative material mask overlay scheme, the material state of a cell is determined by the cumulative effect of a set of circular masks, that is, a cell will be void if the centroid of which is inside a mask. Otherwise, the cell will be solid. In this way, the design variables are defined as the center coordinates and radii of masks, which consecutively determine the density of cells. The influence of the grid size and features (i.e., number and size) of negative masks were investigated in Saxena (2011). To generate structural designs made of bars, Smith and Norato (2020) developed a MATLAB code for the topology optimization of 2D and 3D problems using the geometry projection method. The basic idea behind the geometry projection is to take a high-level parametric description of a given geometric

Journal : Large 158	Article No : 3050	Pages : 54	MS Code : 3050	Dispatch : 28-9-2021

component and map it onto a pseudo-density field over a design region. Accordingly, the projected density ρ at the point $\boldsymbol{x}$ is defined as the volume fraction of the intersection between a ball of radius r centered at $\boldsymbol{x}$ and a geometric component, namely,

$$\rho\left(\frac{\phi_b(\boldsymbol{x}, z_b)}{r}\right) = \begin{cases} 0, & \text{if } \phi_b/r < -1 \\ \tilde{H}(\phi_b/r), & \text{if } -1 \leq \phi_b/r \leq 1 \\ 1, & \text{if } \phi_b/r > 1, \end{cases} \tag{8}$$

where $\phi_b(\boldsymbol{x}, z_b)$ is the signed distance function from $\boldsymbol{x}$ to the boundary of the component, z_b is the vector of geometric parameters that describes the component, and $\tilde{H}(\phi_b/r)$ is a regularized Heaviside function (detailed formulation can be found in their original paper).

Another topology optimization approach using the discrete geometric components is the MMC (and MMB) method. The MMC method proposed by Guo et al. (2014) aims to conduct topology optimization in an explicit and geometrical way using a set of morphable components. The optimized structural topologies are realized by optimizing the geometry characteristic parameters, such as shapes, lengths, thickness, orientations, and layouts of the components. In the MMC-based approach, the structural topology is described in the following way:

$$\begin{cases} \phi^s(\boldsymbol{x}) > 0, & \text{if } \boldsymbol{x} \in \Omega^s \\ \phi^s(\boldsymbol{x}) = 0, & \text{if } \boldsymbol{x} \in \partial\Omega^s \\ \phi^s(\boldsymbol{x}) < 0, & \text{if } \boldsymbol{x} \in D \setminus \Omega^s, \end{cases} \tag{9}$$

where Ω^s denotes a subset of the prescribed design domain D occupied by n components made of solid materials, and $\phi^s(\boldsymbol{x}) = \max_i(\phi_1(\boldsymbol{x}), \dots, \phi_n(\boldsymbol{x}))$ with $\phi_i(\boldsymbol{x})$ being the topology description function representing the geometry of the ith component. Based on the MMC framework, Zhang et al. (2016b) developed a MATLAB code to elaborate the implementation of MMC-based topology optimization using the ersatz material model. Compared with the previous work (Guo et al. 2014), the proposed method is capable of handling components with variable thicknesses by appropriately constructing the topological description functions. Taking advantage of the explicit representation of geometry components, Bai and Zuo (2020) realized the topology optimization of 3D hollow structures via the MMC method, where the hollow components are represented by combining the topology description functions of internal and external components. Recently, Zhao et al. (2021) developed a MATLAB code using MMB to conduct the topology optimization of structures made of 3D hollow bars. In this approach, the geometrical features of the solid bars are first projected onto a fixed grid, where the density of each element can be obtained by a smooth Heaviside approximation of the distance functions. The hollow bars are then constructed by the Boolean subtraction of two solid bars. Based on three existing geometric components approaches, including the geometry projection, MMC, and moving node approaches (Overvelde 2012), Coniglio et al. (2020) proposed a unified approach named generalized geometry projection that unifies the three approaches into one formulation. A saturation strategy is proposed for handling numerical issues encountered during the geometry assembly process. The MATLAB implementation is provided, and effects of parameters on simulation and optimization are discussed. Based on the graph theory, Xing et al. (2021) developed a novel weighted graph representation-based method for the 2D topology optimization problem, where a weighted adjacency matrix is proposed to map the graph property, and an improved differential evolution update scheme with a dual self-adaptive mutation operator (DSADE) is employed as the optimizer. The MATLAB code for this method is available at GitHub.

A summary of the geometric components-based topology optimization methods is shown in Table 9, including the techniques used in topology optimization and FE analysis procedure. Some issues regarding the usability of the codes are also included. Our user experience of the topology optimization codes using geometric components approaches is summarized as follows: (1) Some codes are attached using texts. Users can copy-and-paste the texts to create code files. (2) Some of the research papers only provide kernel functions implementing the proposed method, and readers are recommended to contact the authors to get access to the complete codes.

3 Sizing and ground structure approaches

In this section, we review 8 educational, 6 research, and 1 industrial application papers (with codes) performing structural optimization using sizing and ground structure approaches. Sizing optimization treats member sizes or parameters (e.g., thickness, twist angle, and diameters) as design variables and optimizes structural performances with topology and shape unchanged. The summary of the sizing optimization codes is shown in Table 10. Li and Cao (2016) developed two educational MATLAB codes for reliability analysis and structural optimization using subset simulation, which is a stochastic simulation procedure for estimating small failure probabilities. The design variables are based on problems to be solved, e.g., diameter or the number of coils in a string design as reported in the paper. Lelièvre et al. (2016) investigated the possible robustness and reliability formulations with multiple educational codes provided in Scilab (a free and open-source software for numerical computations, ESI Group 2021) to deal with uncertainties in structural sizing optimization. The design

Journal : Large 158 | Article No : 3050 | Pages : 54 | MS Code : 3050 | Dispatch : 28-9-2021

variables depend on specific problems, e.g., member length and angle of the bracket structure. Jasa et al. (2018) developed an educational open-source program (`OpenAeroStruct`) within the OpenMDAO framework (Gray et al. 2019), handling low-fidelity aerostructural analysis and sizing optimization of design variables including twist distribution, spar thickness, and platform variables for aerostructures. Huang et al. (2019) developed an evidence-theory-based design optimization considering parametric correlations and provided a corresponding MATLAB code, which is an effective computational tool for the structural reliability design involving epistemic uncertainties. The design variables are determined by the problems on hand, e.g., height and width of cross sections for a cantilever beam. Belotti et al. (2021) proposed a multi-domain approach to optimize the dynamic response of vibrating linear systems under actuation by varying the values of lumped masses. The corresponding MATLAB implementations are provided. Inspired by Pareto's principle, Shaqfa and Beyer (2021) proposed a global optimization approach and explored its capabilities using 26 standard benchmark examples with design variables depending on specific problems. Numerical implementations in C++, Python, and Octave (MATLAB) are provided for different users. Using kinematic models to approximate the behavior of fabrics, Krogh et al. (2021) conducted simulation and optimization of the draping of a composite material fabric onto a mold. The design variables are the origin point and initial draping direction. Both MATLAB and Python codes are provided for educational use. Ning (2021) performed design optimization of wind turbine blades using blade element momentum methods (BEM) and gradient-based update scheme. Design variables include the blade chord distribution, twist distribution, tip-speed ratio, and the pitch at 80 wind speeds from the cut-in to the cut-out wind speeds. The code implemented in Julia programming language is provided. We note that some of the reviewed papers in sizing optimization adopted relatively general optimization formulations, which require modifications to the specific design problems at hand.

The ground structure method (GSM) treats member sizes, such as the cross-sectional areas of structural members (e.g., trusses and beams), as design variables. In addition, because some studies of GSM also optimize the structural connectivity and layout by completely removing (and adding) members from the initial ground structure (e.g., Zhang et al. 2017), GSM is sometimes categorized as a topology optimization approach. Thus, we separate the GSM from the other sizing optimization approaches and review it as an independent sub-category. The basic optimization formulations (i.e., elastic and plastic design formulations) of GSM to design minimum volume truss can be given as (Zegard and Paulino 2014; Bendsøe and Sigmund 2013):

$$
\begin{aligned}
&\min_{\mathbf{a}} : \; V = \mathbf{l}^T \mathbf{a} \\
&\quad \text{s.t.} \;:\; -\sigma_C \le \sigma_i(\mathbf{a}) \le \sigma_T \quad \text{if } a_i > 0 \\
&\quad a_i \ge 0 \quad i = 1, 2, \dots, N_t \\
&\quad \mathbf{K}\mathbf{u}(\mathbf{a}) = \mathbf{F},
\end{aligned}
\tag{10}
$$

or the limit design form, also termed plastic design, that only requires force equilibrium (i.e., without kinematic compatibility)

$$
\begin{aligned}
&\min_{\mathbf{a}} : \; V = \mathbf{l}^T \mathbf{a} \\
&\quad \text{s.t.} \;:\; -\sigma_C a_i \le n_i(\mathbf{a}) \le \sigma_T a_i \quad i = 1, 2, \dots, N_t \\
&\quad \mathbf{B}^{\mathrm{T}}\mathbf{n}(\mathbf{a}) = \mathbf{F},
\end{aligned}
\tag{11}
$$

respectively. In the formulations, N_t is the number of truss members with index i denoting the ith truss member; V is the total volume obtained from the dot product of the length vector $\mathbf{l}$ and the cross-section area vector $\mathbf{a}$ (which is also the design variable) of the truss. In the equilibriums, $\mathbf{F}$ is the nodal force vector; $\mathbf{K}$ and $\mathbf{B}$ are global stiffness and nodal equilibrium matrices, respectively; $\mathbf{u}$ and $\mathbf{n}$ are the displacement and axial force vectors, respectively. In the constraints, σ_i, σ_C, and σ_T are evaluated stress of ith element, compression stress limit, and tension stress limit, respectively. In some studies (e.g., Stolpe 2019), the elastic formulation can be transformed to the one of compliance minimization with a volume constraint. We note that the reviewed papers, summarized in Table 11, use either of the two formulations (i.e., elastic or plastic design formulations).

With educational purposes, Sokół (2011) developed a 99-line GSM code implemented in Mathematica (Wolfram 2021), generating least-weight trusses based on linear programming. Zegard and Paulino (2014, 2015) developed GSM to obtain least-weight trusses in both 2D and 3D and enabled the flexible definition of restriction zones (i.e., geometric entities that no bar should intersect), which allows for the use of GSM with arbitrary (in particular concave) domain geometries. The corresponding MATLAB codes named as `GRAND` (GRound structure ANalysis and Design) and `GRAND3` (GRound structure ANalysis and Design in 3D), respectively, are provided for educational and research purposes. He et al. (2019) developed an educational 98-line Python script adopting the adaptive "member adding" scheme for efficiently solving 2D truss layout optimization problems considering multiple load cases, joint costs, and nonconvex domains. Stolpe (2019) tackled the optimization of fail-safe performance using the GSM and provided main CVX codes (CVX is a MATLAB-based modeling system for convex optimization). Based on the elastic formulation of GSM (i.e., compliance minimization), Kanno (2020)

Journal : Large 158 | Article No : 3050 | Pages : 54 | MS Code : 3050 | Dispatch : 28-9-2021

Table 9 Summary of geometric components/bars topology optimization codes

Type	Name/Reference	Environment	Summary	Techniques: Topology optimization	Techniques: FE method	Usability
Research	"*mmos*" Saxena A (2011) https://doi.org/10.1007/s00158-011-0649-4	MATLAB	– 2D – Compliance minimization; displacement maximization – Modeling topology optimization via negative masks to obtain continua using gradient search	– No filtering technique – No material interpolation – Update scheme: Material Mask Overlay Scheme (MMOS)	– Linear elasticity – Direct solver – Unit Wachspress hexagonal discretization	– Code attached using texts (users can copy-and-paste the texts to create code files) – Encounter syntax error when using the attached code
Educational	"*GPTO*" Smith H, Norato JA (2020) https://doi.org/10.1007/s00158-020-02552-0	MATLAB	– 2D and 3D – Compliance minimization – A MATLAB code to perform topology optimization of 2D and 3D structures made of cylindrical bars using geometry projection	– Regularized Heaviside projection – SIMP (or RAMP) interpolation – MATLAB function "*fmincon*" or MMA update scheme	– Linear elasticity – Direct solver; iterative solver; use of the GPU card to solve the system of linear equations – Mesh generation options: '*generate*'; '*read-home-made*'; '*read-gmsh*'	– Code available at https://github.com/jnorato/GPTO – Require "*mmasub.m*" and "*subsolv.m*" – A reference manual with a step-by-step tutorial is provided to reproduce the first example in this paper
Research	"*MMC188*" Zhang W, Yuan J, Zhang J, Guo X (2016) https://doi.org/10.1007/s00158-015-1372-3	MATLAB	– 2D – Compliance minimization – A new topology optimization approach based on the moving morphable components (MMC) method	– No filtering technique – Ersatz material model – MMA update scheme	– Linear elasticity – Direct solver – Quadrilateral discretization	– Code attached using texts (users can copy-and-paste the texts to create code files) – Require "*mmasub.m*" and "*subsolv.m*"
Research	"*Hollow MMC*" Bai J, Zuo W (2020) https://doi.org/10.1007/s00158-019-02353-0	MATLAB	– 3D – Compliance minimization – An MMC method to optimize 3D hollow structures	– No filtering technique – Ersatz material model – MMA update scheme	– Linear elasticity – Direct solver – Hexahedral discretization	– Code attached using ESM – Partial codes provided
Research	"*MMB_3D*" Zhao Y, Hoang VN, Jang GW, Zuo W (2021) https://doi.org/10.1016/j.advengsoft.2020.102955	MATLAB	– 3D – Compliance minimization – A moving morphable bars (MMB) method to optimize 3D hollow structures	– No filtering technique – Material interpolation similar to modified SIMP method – MMA update scheme	– Linear elasticity – Direct solver – Hexahedral discretization	– Code attached using texts (users can copy-and-paste the texts to create code files) – Require "*mmasub.m*" and "*subsolv.m*" – Partially based on top3d
Research	"*GGP*" Coniglio S, Morlier J, Gogu C,Amargier R (2020) https://doi.org/10.1007/s11831-019-09362-8	MATLAB (Octave incompatible: '*replace*' undefined)	– 2D – Compliance minimization – Generalized geometry projection (GGP): A unified geometric component topology optimization method for geometry projection, MMC, and moving node approaches	– No filtering technique – Geometric assembly: Saturation strategy – MMA update scheme	– Linear elasticity – Direct solver – Quadrilateral discretization	– Code available at https://github.com/topggp/GGP-Matlab – Ready to use – Built upon top88 – Discuss effects of GGP parameters on simulation and optimization

Journal : Large 158 | Article No : 3050 | Pages : 54 | MS Code : 3050 | Dispatch : 28-9-2021

Table 9 (continued)

Type	Name/Reference	Environment	Summary	Techniques		Usability
				Topology optimization	FE method	
Research	"*WGM*" Xing J, Xu P, Yao S, Zhao H, Zhao Z, Wang Z (2021) https://doi.org/10.1016/j.advengsoft.2021.102977	MATLAB (Octave compatible)	– 2D – Compliance minimization – A weighted graph representation-based method (WGM) for topology optimization	– No filtering technique – Weighted adjacency matrix – Differential evolution update scheme with a dual self-adaptive mutation operator (DSADE)	– Linear elasticity – Direct solver – Quadrilateral discretization	– Code available at https://github.com/CSUxingjie/WGMAlgorithm – Ready to use – Partially based on top99

investigated three approaches for robust design optimization, i.e., worst-case optimization, discrepancy minimization (namely, minimizing the gap between the worst-case and nominal compliance values), and variance minimization. An educational MATLAB code is provided in this paper. Adopting GSM and Wang tiling assembly formalism, Tyburec et al. (2021) performed a concurrent optimization of both truss modules topologies and their macroscopic assembly. A MATLAB code is provided for result reproduction. We note that, in addition to having different educational/research purposes, many of the reviewed papers using GSM cast their formulations as linear or semidefinite programming problems, which can be solved efficiently using existing optimization tools.

To conclude this section, we summarize the user experience for the codes of sizing and ground structure approaches: (1) The sizing optimization codes are typically independent of each other in terms of target problems, employed algorithms, and implementation environment and style. (2) Most codes attached in the papers using sizing and ground structure approaches are directly downloadable (e.g., attached as ESM) and ready-to-use.

4 Shape optimization

The third category of SMO is shape optimization, which refers to optimizing the structural shape by only varying the boundary of the structural domain (i.e., no hole is created or removed from the structural domain). In this section, 5 educational and 1 research papers for shape optimization problems are reviewed, which are compiled in Table 12. In order to solve the governed PDEs of the state equations efficiently, most of those papers leverage existing open-source FE software, which can reduce the computational effort for FE analysis and shape sensitivity analysis. Allaire and Pantz (2006) demonstrated shape optimization routines for two classical methods, the boundary variation method and the homogenization method using FreeFem++. The compliance minimization and gripper optimization problems were exemplified in their work. It is motivated that the proposed routines can be assigned to graduate students as numerical homework to motivate the understanding of shape optimization. Dapogny et al. (2018) developed a FreeFem++ code for Navier–Stokes fluid design problems using shape optimization, aiming to either minimize the dissipated energy or achieve a targeted velocity profile. Gangl et al. (2021) conducted the shape optimization using the FE software package NGSolve (Schöberl 2014), which can solve a large number of boundary value problems efficiently, considering both unconstrained and PDE-constrained cases. Both semi-automatic and fully automatic approaches for calculating the first- and second-order shape derivatives are presented

Journal : Large 158 | Article No : 3050 | Pages : 54 | MS Code : 3050 | Dispatch : 28-9-2021

Table 10 Summary of sizing optimization codes

Type	Name/Reference	Environment	Summary	Techniques		Usability
				Optimization	FE method	
Educational	Li HS, Cao ZJ (2016) https://doi.org/10.1007/s00158-016-1414-5	MATLAB (Octave compatible)	– 2D – Generic objective function – Design variables depend on specific problems (e.g., wire diameter, mean coil diameter, and the number of active coils of a string design) – Reliability analysis and structural optimization based on subset simulation	– Subset simulation – Sample and update based on Markov chain Monte Carlo	– Linear equilibrium – Direct solver – Frame or spring element	– Code available at https://sites.google.com/site/rasosubsim/ – Ready to use
Educational	Lelievre N, Beaurepaire P, Mattrand C, Gayton N, Otsmane A (2016) https://doi.org/10.1007/s00158-016-1556-5	Scilab	– 2D – Generic objective function – Design variables depend on specific problems (e.g., member length or angle of a bracket structure) – Investigate possible sizing optimization formulations incorporating robustness and reliability	– Signal-to-noise-ratio to measure the robustness – Monte Carlo simulation – Update scheme based on Nelder-Mead algorithm	– N/A (analytical solution)	– Code attached using ESM
Educational	"*OpenAeroStruct*" Jasa JP, Hwang JT, Martins JRRA (2018) https://doi.org/10.1007/s00158-018-1912-8	Python	– 1D – Generic objective function – Design variables are twist distribution, spar thickness distribution, and platform variables – Develop a low-fidelity aerostructural analysis and optimization tool within the OpenMDAO framework	– Coupled adjoint method to compute derivatives – Breguet range equation to compute the fuel burn – Update scheme based on sequential quadratic programming	– Linear equilibrium – Direct solver – Vortex lattice method – Beam elements	– Code available at https://github.com/mdolab/openaerostruct

Journal : Large 158 | Article No : 3050 | Pages : 54 | MS Code : 3050 | Dispatch : 28-9-2021

Table 10 (continued)

Type	Name/Reference	Environment	Summary	Techniques		Usability
				Optimization	FE method	
Research	Huang ZL, Jiang C, Zhang Z, Zhang W, Yang TG (2019) https://doi.org/10.1007/s00158-019-02225-7	MATLAB (Octave incompatible: Syntax error)	– 1D, 2D, or 3D – Generic objective function – Design variables depend on specific problems (e.g., height and width of a beam cross-section) – Evidence-theory-based optimization to design structures involving epistemic uncertainties	– Evidence-theory-based reliability analysis – Monte Carlo simulation – Sequential quadratic programming	– Linear equilibrium – Direct solver – Hexahedron element	– Code attached using ESM – Ready to use
Industrial application	Belotti R, Richiedei D, Trevisani A (2021) 10.1007/s00158-020-02709-x	MATLAB	– 1D – Rank minimization – Design variables are the values of the lumped mass – Multi-domain optimization of the dynamic response of an underactuated vibrating linear system	– Semidefinite embedding lemma to solve the rank-minimization optimization problem	– Linear time-invariant equilibrium – Direct solver – Beam element	– Code attached using images – Need access to YALMIP
Research	Shaqfa M, Beyer K (2021) https://doi.org/10.1007/s00500-021-05853-8	Python, MATLAB, and C++ (Octave compatible)	– 2D – Generic objective function – Design variables depend on specific problems (e.g., cross-section areas of truss members) – Propose a simple global optimization algorithm inspired by Pareto's principle	– Pareto-like sequential sampling – Monte Carlo sampling	– N/A (analytical solution)	– Code available at https://github.com/eesd-epfl/pareto-optimizer – Ready to use

Journal : Large 158 | Article No : 3050 | Pages : 54 | MS Code : 3050 | Dispatch : 28-9-2021

Table 10 (continued)

Type	Name/Reference	Environment	Summary	Techniques		Usability
				Optimization	FE method	
Educational	"*KinDrape*" Krogh C, Bak BL, Lindgaard E, Olesen AM, Hermansen SM, Broberg PH, Kepler JA, Lund E, Jakobsen J (2021) https://doi.org/10.1007/s00158-021-02925-z	MATLAB and Python (Octave incompatible: Requires "*rng*" function)	– 2D – Minimization of the shear angles of the fabric on the mold – Design variables are the origin point and initial draping direction – Simulate and optimize the draping of a composite material fabric onto a mold	– The kinematic draping algorithm – Genetic algorithm	– N/A (a kinematic model is established)	– Code available at https://doi.org/10.5281/zenodo.4316860 – Ready to use
Educational	Ning A (2021) https://doi.org/10.1007/s00158-021-02883-6	Julia	– 1D and 2D – Maximization of the annual energy production – Design variables are the blade chord distribution, twist distribution, tip-speed ratio, and the pitch at 80 wind speeds from the cut-in wind speed to the cut-out wind speed – Design optimization of blades using blade element momentum methods with guaranteed convergence and machine precision	– Blade element momentum methods – Graph coloring technique – Sequential quadratic programming	– N/A (blade element momentum methods are adopted)	– Code available at https://github.com/byuflowlab/ning2020-bem

Journal : Large 158 | Article No : 3050 | Pages : 54 | MS Code : 3050 | Dispatch : 28-9-2021

Table 11 Summary of ground structure method codes

Type	Name/Reference	Environment	Summary	Techniques: Optimization/others	Techniques: FE method	Usability
Research	SokółT (2011) https://doi.org/10.1007/s00158-010-0557-z	Mathematica	– 2D – Volume minimization using plastic design formulation – GSM to design least-weight truss	– Update scheme based on linear programming (e.g.,interior point method)	– Linear equilibrium – Direct solver – Truss element	– Code attached using ESM
Educational	"*GRAND*" Zegard T, Paulino GH (2014) https://doi.org/10.1007/s00158-014-1085-z	MATLAB (Octave compatible)	– 2D – Volume minimization using plastic design formulation – GSM to design least-weight trusses with restriction zones	– Enable restriction zones using collision detection algorithms – Collinearity check for the generated ground structure – Update scheme based on linear programming (e.g., interior point method)	– Linear equilibrium – Direct solver – Truss element	– Code attached using ESM – Ready to use
Research	"*GRAND3*" Zegard T, Paulino GH (2015) https://doi.org/10.1007/s00158-015-1284-2	MATLAB (Octave compatible)	– 3D – Volume minimization using plastic design formulation – GSM to design least-weight trusses with restriction zones	– Enable restriction zones using collision detection algorithms – Collinearity check for the generated ground structure – Update scheme based on linear programming (e.g., interior point method)	– Linear equilibrium – Direct solver – Truss element	– Code attached using ESM – Ready to use
Educational	He L, Gilbert M, Song X (2019) https://doi.org/10.1007/s00158-019-02226-6	Python	– 2D – Volume minimization using plastic design formulation – GSM to design least-weight trusses with adaptive "member adding" scheme	– Adaptive "member adding" scheme based on "column generation' approach – Notional joint cost penalizing short members – Update scheme based on linear programming (e.g., interior point method)	– Linear equilibrium – Direct solver – Truss element	– Code attached using ESM
Research	Stolpe M (2019) https://doi.org/10.1007/s00158-019-02295-7	MATLAB	– 2D – Compliance minimization using elastic design formulation – GSM to design fail-safe trusses using the working-set algorithm	– Member damage or degradation failure model – Working-set algorithm based on solving a sequence of convex relaxations – Update scheme based on semidefinite programming (e.g., interior point method)	– Linear equilibrium – Direct solver – Truss element	– Code attached using texts (users can copy-and-paste the texts to create code files) – Partial CVX codes provided

Journal : Large 158 | Article No : 3050 | Pages : 54 | MS Code : 3050 | Dispatch : 28-9-2021

Table 11 (continued)

Type	Name/Reference	Environment	Summary	Techniques		Usability
				Optimization/others	FE method	
Educational	Kanno Y (2020) https://doi.org/10.1007/s00158-020-02503-9	MATLAB (Octave incompatible: Syntax error)	– 2D – Worst-case compliance minimization; the discrepancy minimization; the variance minimization (using elastic design formulation) – Investigate the approaches of robust truss optimization using GSM	– Difference-of-convex algorithm – Update scheme based on semidefinite programming (e.g., interior point method)	– Linear equilibrium – Direct solver – Truss element	– Code available at https://github.com/ykanno22/relative_robust/ – Ready to use
Research	Tyburec M, Zeman J, Doskar M, Kruzık M, Leps M (2021) https://doi.org/10.1007/s00158-020-02744-8	MATLAB	– 2D – Compliance minimization for truss using elastic design formulation; minimization of the weighted average of the complementary strain energies for multiple load cases – Concurrent design truss modules topologies (using GSM) and their macroscopic assembly	– Wang tiling formalism to encode macroscopic assembly – Bilevel optimization strategy – Second-order cone programming for the lower-level truss design problem – Genetic algorithm for the upper-level assembly problem	– Linear equilibrium – Direct solver – Truss element	– Code available at https://doi.org/10.5281/zenodo.3835555 – Need access to YALMIP

Journal : Large 158 | Article No : 3050 | Pages : 54 | MS Code : 3050 | Dispatch : 28-9-2021

Table 12 Summary of shape optimization codes

Type	Name/Reference	Environment	Summary	Techniques: Optimization	Techniques: FE method	Usability
Educational	Allaire G, Pantz O (2006) https://doi.org/10.1007/s00158-006-0017-y	FreeFem++	– 2D – Compliance minimization; maximization of the pressure of the grip on the piece – Showcase shape optimization routines using the FreeFem++	– Boundary variation method – Homogenization method	– Linear elasticity – PDE solver in FreeFem++ – Triangular discretization	– Code available at http://www.cmap.polytechnique.fr/~optopo – Need access to FreeFem++
Educational	Dapogny C, Frey P, Omnes F, Privat Y (2018) https://doi.org/10.1007/s00158-018-2023-2	FreeFem++	– 2D – Minimization of the dissipated energy; minimization of the discrepancy with a reference – FreeFEM++ code to perform shape optimization of Navier–Stokes flow problem	– Hadamard boundary variation method for calculating the sensitivity	– Navier–Stokes flow – Iterative solver – Triangular discretization	– Code available at https://github.com/flomnes/optiflow – Need access to FreeFem++
Educational	Gangl P, Sturm K, Neunteufel M, Schoberl J (2021) https://doi.org/10.1007/s00158-020-02742-w	NGSolve	– 2D and 3D – Generic objective function – Showcase how to obtain first– and second-order shape derivatives for unconstrained and PDE-constrained shape optimization problems using NGSolve	– Semi-automatic shape differentiation – Fully automated shape differentiation	– Nonlinear elasticity; Maxwell's equations; Helmholtz's equation – PDE solver in NGSolve – Triangular and tetrahedral discretization	– Code attached using ESM – Need access to NGSolve
Educational	"*OpenFEMflow*" Elham A, van Tooren MJ (2021) https://doi.org/10.1007/s00158-020-02799-7	MATLAB (Octave incompatible: *'feature'* undefined)	– 2D – Generic objective function – Discrete adjoint aerodynamic shape optimization based on symbolic analysis	– Symbolic analysis	– Computational fluid dynamics – Iterative solver – Triangular discretization	– Code available at https://github.com/mdotubs/OpenFEMflow – Ready to use
Educational	Paganini A, Wechsung F (2021) https://doi.org/10.1007/s00158-020-02813-y	Fireshape	– 2D and 3D – Compliance minimization; minimization of the kinetic energy dissipation into heat of a pipe – An automated shape optimization toolbox for Firedrake	– Moving mesh method – Compute adjoint equations and shape derivatives in an automated fashion	– Linear elasticity; nonlinear Navier–Stokes equations – Generate the mesh using '*Gmsh*' – Use solvers and preconditioners accessible from PETSc	– Fireshape available at https://github.com/Fireshape/Fireshape – Need access to Fireshape, Firedrake, Gmsh, Rapid Optimization Library, and PETSc

Journal : Large 158 | Article No : 3050 | Pages : 54 | MS Code : 3050 | Dispatch : 28-9-2021

Table 12 (continued)

Type	Name/Reference	Environment	Summary	Techniques		Usability
				Optimization	FE method	
Research	Ghantasala A, Asl RN, Geiser A, Brodie A, Papoutsis E, Bletzinger KU (2021) https://doi.org/10.1007/s10957-021-01826-x	C++ and Python	– 2D and 3D – Generic objective function – A framework for simulation-based large-scale shape optimization using vertex morphing	– Constrained node-based shape optimization using vertex morphing technique	– Offer detached interface to use external solvers as black-box	– Code (KratosMultiphysics) available at https://github.com/KratosMultiphysics/Kratos

in that work. Elham and van Tooren (2021) performed aerodynamic shape optimization with computational fluid dynamics simulation based on symbolic analysis and provided a MATLAB code named as `OpenFEMflow`. Paganini and Wechsung (2021) introduced an open-source shape optimization toolbox (`Fireshape`) built upon the FE software Firedrake (Rathgeber et al. 2016), which is capable of calculating the shape derivatives automatically. The `Fireshape` also allows for the access to PETSc and Rapid Optimization Library (ROL) to employ their solvers and optimization algorithms. Another notable shape optimization approach termed vertex morphing has been developed by Bletzinger (2014). Using the proposed vertex morphing approach, Ghantasala et al. (2021) developed a framework for simulation-based large-scale shape optimization. This approach has the following characteristics: (a) it mirrors topology optimization in that all boundary grids move as independent variables; (b) a filtering function very similar to that of topology optimization is utilized to guarantee a smooth and spline-like boundary shape representation. Their research work is implemented in the large open-source code `KratosMultiphysics`. An alternative filter formulation was presented by Zhou et al. (2018) for the shape optimization of fluid flow problems.

5 Building blocks for SMO methods

Apart from the development of an integrated topology optimization framework, a number of papers (8 educational papers, 5 research papers, 1 review paper, and 1 original software publication) contribute to establishing useful building blocks for various SMO methods such as mesh generator, FE modeling, design update scheme, filtering, and post-processing. A brief summary of those building blocks can be found in Table 13.

For the mesh generator, Talischi et al. (2012a) developed a MATLAB code, named `PolyMesher`, to generate polygonal meshes for arbitrary domain geometries. `PolyMesher` is later used in the code `PolyTop` (Talischi et al. 2012b) as the mesh generator. For the FE modeling, Andreassen and Andreasen (2014) developed a self-contained MATLAB code on how to determine the effective properties of 2D composite materials using the numerical homogenization method. Subsequently, Dong et al. (2019) developed an educational homogenization code written in MATLAB for 3D cellular materials. For thermal problems, Beckers and Beckers (2015) developed an educational MATLAB code for performing dual analysis of heat conduction problems. Tauzowski et al. (2019) introduced a programming concept, the function object (termed functor), for the FE implementation in topology optimization problems considering elasto-plastic materials. Instead of

Journal : Large 158 | Article No : 3050 | Pages : 54 | MS Code : 3050 | Dispatch : 28-9-2021

using the traditional FE method, Gao et al. (2021) developed a MATLAB framework implementing the isogeometric topology optimization method proposed by Gao et al. (2019a). Note that although an integrated framework is provided, we categorize it into the building blocks section as the main contribution lies in the isogeometric analysis. For the optimization procedure, Dzierzanowski (2012) derived formulas of the optimal material distribution for the compliance minimization problem considering various material interpolation schemes, and developed MATLAB codes based on corresponding exact solutions. For design update schemes, Kumar and Suresh (2021) replaced the bisection method with the direct method to compute the Lagrange multiplier in the OC algorithm. A drop-in MATLAB implementation of the direct method is provided in the paper, which can be directly plugged into other topology optimization codes. For the sensitivity analysis, Chandrasekhar et al. (2021) developed a framework named `AuTO` to implement automatic differentiation in topology optimization by employing JAX, a Python library to compute sensitivities automatically. The usability and advantage of the `AuTO` framework are demonstrated by three standard density-based problems, i.e., compliance minimization, compliant mechanism, and material design. To achieve a black-and-white design, Sigmund and Maute (2013) provided a drop-in MATLAB threshold code snippet based on `top99` and `top88` to map the gray-scale design obtained from SIMP codes to a discrete design that satisfies the volume constraint. Huang (2021) incorporated the floating projection constraint into topology optimization to seek a smooth or black-and-white design employing the ersatz material model or a material penalization model. For the filtering and post-processing, Langelaar (2017) developed an additive manufacturing filter, `AMfilter`, which can be incorporated into the density-based topology optimization to generate print-ready designs without additional supports. A 2D MATLAB code implementing the proposed filter and guidelines for integrating it into `top88` are provided. Zhang et al. (2017) proposed a discrete filter scheme for the GSM, which can be applied to 2D and 3D truss optimization to facilitate manufacturability, allow for the definition of valid structure, and achieve reduced-order modeling in both the state and optimization problems. A MATLAB implementation of the proposed filter operator is provided. To bridge topology optimization and additive manufacturing, Zegard and Paulino (2016) developed a streamlined procedure for generating additive-manufacturing-ready file formats (STL, or stereolithography) from topology optimized designs. Specifically, a graphical tool (`TOPslicer`) for the 3D density-based topology optimization is provided as ESM. Recently, Ibhadode et al. (2021) developed a framework, `IbIPP`, in MATLAB to perform 2D topology optimization from initialization to post-processing. The employment of an image-based initialization makes it capable of considering arbitrary domains, and the post-processing function can generate STL file readily for additive manufacturing. For the optimization subroutine, adjusted MATLAB codes (i.e., `top88`, `esoL`, and `levelset88`) for SIMP (Andreassen et al. 2011), BESO (Xia et al. 2018), and level-set approaches (Otomori et al. 2015) are provided to accommodate the `IbIPP` framework, respectively.

6 Papers with educational values (without codes)

In addition to articles that provide codes, many educational papers focus on aspects related to teaching, fundamental concepts, and interactive applications, which bring invaluable contributions to the community. This section reviews papers within the latter category. Three educational articles in structural design and optimization are developed to explain the classroom teaching experience and suggestions. Haftka and Jenkins (1998) described the experience of a classroom project about maximizing the tension strength of a riveted lap joint based on both analytical and experimental structural optimization. Filomeno Coelho et al. (2014) presented the project-based learning for form-finding and structural optimization by describing the teaching experience of a graduate student course and provided guidelines for developing project-based courses in structural optimization. Sangree et al. (2015) discussed their efforts in leveraging topology optimization as a teaching tool and incorporating it into undergraduate courses to inspire structural design creativity. Meanwhile, 4 educational papers aim to explain and discuss fundamental and critical concepts for structural optimization problems. Stolpe (2010) provided illustrative examples to discuss some fundamental properties (e.g., uniqueness of solutions) of structural topology optimization. Klarbring (2015) developed a unified structural optimization framework using state problem functionals as objective. This paper starts with the master state problem (i.e., the canonical equations) and then discussed special cases including linear elasticity, Darcy–Stokes flow, and pipe flow problems. In terms of shape optimization, Wang and Kumar (2017) investigated the transient heat conduction problem using isogeometric analysis, and introduced the numerical implementation of a continuous adjoint method to conduct the shape sensitivity analysis. In order to complete the learning experience for students and to facilitate classroom teaching, Zhou and Sigmund (2021) provided complementary lecture notes focusing on the theoretical foundation of `top99`/`top88` codes with self-contained content from several aspects, including OC update scheme, closed-form

Journal : Large 158 | Article No : 3050 | Pages : 54 | MS Code : 3050 | Dispatch : 28-9-2021

Table 13 Summary of papers that provide building block codes in SMO

Type	Name/Reference	Environment	Summary	Usability
Educational	"*PolyMesher*" Talischi C, Paulino GH, Pereira A, Menezes IFM (2012a) https://doi.org/10.1007/s00158-011-0706-z	MATLAB (Octave compatible)	– 2D – Pre-process: PolyMesher (mesh generator for polygonal elements)	– Code attached using ESM – Ready to use – This mesh generator can also be used to generate certain uniform meshes (regular tessellations)
Educational	"*homogenize*" Andreassen E, Andreasen CS (2014) https://doi.org/10.1016/j.commatsci.2013.09.006	MATLAB (Octave compatible)	– 2D – Modeling/FE analysis: To determine the effective macroscopic properties of a periodic two-material composite using numerical homogenization	– Code attached using ESM – Ready to use
Educational	"*homo3D*" Dong G, Tang Y, Zhao YF (2019) https://doi.org/10.1115/1.4040555	MATLAB (Octave incompatible: *fgetl: invalid stream number = -1*)	– 3D – Modeling/FE analysis: A numerical homogenization method for 3D cellular materials	– Code available at https://github.com/GuoyingDong/homogenization – Ready to use
Educational	"*Dual_66*" Beckers P, Beckers B (2015) https://doi.org/10.1016/j.camwa.2015.09.007	MATLAB	– 2D – Modeling/FE analysis: A compact MATLAB implementation of a finite element code performing dual analysis of heat conduction problems	– Code attached using images – Built upon top99
Research	Tauzowski P, Blachowski B, Lógó J (2019) https://doi.org/10.1016/j.advengsoft.2019.102690	C++	– 2D – Modeling/FE analysis: Functor-oriented approach to FE programming for topology optimization of elasto-plastic structures	– Code attached using images (a list of coding examples to show the functor-oriented approach to FE programming)
Educational	"*IgaTop*" Gao J, Wang L, Luo Z, Gao L (2021) https://doi.org/10.1007/s00158-021-02858-7	MATLAB (Octave incompatible: Requires NURBS toolbox)	– 2D – Modeling/FE analysis: An integrated MATLAB code implementing the isogeometric analysis into the topology optimization	– Code attached using ESM – Need access to NURBS toolbox
Research	"*tophomog4*" and "*topgramp1*" Dzierżanowski G (2012) https://doi.org/10.1007/s00158-012-0788-2	MATLAB (Octave compatible)	– 2D – Optimization: Derive explicit formulae of material distribution to the compliance minimization problem for various interpolation schemes	– Code attached using texts (users can copy-and-paste the texts to create code files) – Ready to use – Built upon top88
Educational	Kumar T, Suresh K (2021) https://doi.org/10.1007/s00158-020-02740-y	MATLAB	– 2D and 3D – Optimization: A direct method for computing Lagrange multiplier in OC update scheme	– Code attached using images – Provide drop-in replacements in top99, top88, top3d, volume minimization code and PolyMat
Educational	"*AuTO*" Chandrasekhar A, Sridhara S, Suresh K (2021) http://arxiv.org/abs/2104.01965	JAX (Python)	– Optimization/Sensitivity analysis: Automatic differentiation framework in topology optimization (AuTO)	– Code available at https://github.com/UW-ERSL/AuTO – Ready to use

Journal : Large 158 | Article No : 3050 | Pages : 54 | MS Code : 3050 | Dispatch : 28-9-2021

Table 13 (continued)

Type	Name/Reference	Environment	Summary	Usability
Review	"*Threshold code*" Sigmund O, Maute K (2013) https://doi.org/10.1007/s00158-013-0978-6	MATLAB (Octave compatible)	– 2D – Post-processing: Convert a gray-scale design obtained with SIMP codes to a discrete design satisfying the volume constraint	– Code attached using images – Provide drop-in codes in top99 and top88
Research	"*FPTO2D*" Huang X (2021) https://doi.org/10.1016/j.advengsoft.2020.102942	MATLAB (Octave incompatible: *'true-size' undefined*)	– 2D – Optimization/Post-processing: A floating projection topology optimization (FPTO) method with smooth boundary representation using the ersatz material model	– Code attached using ESM – Ready to use – Built upon top88
Research	"*AMfilter*" Langelaar M (2017) https://doi.org/10.1007/s00158-016-1522-2	MATLAB (Octave compatible)	– 2D – Filtering/Post-processing: An additive manufacturing filter for the density-based topology optimization to generate print-ready designs	– Code attached using ESM – Provide step-by-step guidelines for integrating the proposed filter into top88
Research	"*Discrete filter*" Zhang X, Ramos AS, Paulino GH (2017) https://doi.org/10.1007/s00158-016-1627-7	MATLAB	– 2D and 3D – Filtering/Post-processing: A discrete filter for reduced-order modeling of 2D and 3D truss optimization (GSM) considering multiple load cases and nonlinear material behavior	– Code attached using images – Provide the MATLAB implementation of the discrete filter function
Educational	"*TOPslicer*" Zegard T, Paulino GH (2016) https://doi.org/10.1007/s00158-015-1274-4	MATLAB (Octave incompatible: *rotate3d: invalid figure handle HFIG*)	– 2D and 3D – Post-processing: Introduce a procedure bridging TO and additive manufacturing (TOPslicer)	– Code attached using ESM – Ready to use
Original software publication	"*IbIPP*" Ibhadode O, Zhang Z, Bonakdar A, Toyserkani E (2021) https://doi.org/10.1016/j.softx.2021.100701	MATLAB (Octave incompatible: *Magick++ exception*)	– 2D – Initialization and post-processing: A framework to perform 2D topology optimization covering from image-based initialization to data post-processing for additive manufacturing	– Code available at https://github.com/ElsevierSoftwareX/SOFTX-D-21-00033 – Ready to use – Incorporate adjusted open-source MATLAB codes, including top88, esoL, and levelset88 into IbIPP

Journal : Large 158 | Article No : 3050 | Pages : 54 | MS Code : 3050 | Dispatch : 28-9-2021

update scheme for Lagrange multiplier, and a derivation of the compliance sensitivity.

Finally, based on the development of different structural optimization approaches, 7 interactive applications have been developed by researchers creating auxiliary educational tools for beginners and classroom teaching. The first application is a web-based topology optimization program developed by Tcherniak and Sigmund (2001) to elucidate the basic concepts and ideas as well as to serve as a computer-aided learning tool for students. Then, the `TopOpt` app solving the 2D compliance problem was released by Aage et al. (2013), which can be used on both desktop computers and handheld devices. The underlying code is inspired by the publicly available 88 and 99-line MATLAB codes (Sigmund 2001; Andreassen et al. 2011). A 3D version named `TopOpt 3D` app was developed (Nobel-Jørgensen et al. 2015), targeting both desktop computers and handheld devices. An educational game, `TopOpt Game`, which is designed for users to solve the 2D compliance minimization topology optimization problem, was developed by Nobel-Jørgensen et al. (2016). By gamifying topology optimization, the overall concepts are introduced in a new way for students. Nguyen et al. (2020) developed an efficient hybrid method for structural optimization, where the topology is first estimated using the density representation, then the result is utilized as an initialization of the subsequent shape optimization. Following the proposed method, an app named `TopOpt Shape` was developed. For the layout optimization of trusses, Fairclough et al. (2021) developed an interactive real-time web app named `LayOpt` that can be used on various computing devices to optimize the topology of 2D trusses. To expand the involvement of SMO toward architectural engineers and architects, an add-on for Grasshopper is developed, `Millipede`, which can conduct topology optimization on various structural systems and visualize the final results.

7 Numerical assessments

In this section, we conduct numerical experiments using the benchmark cantilever beam example on 10 MATLAB codes that solve the standard minimum compliance problem, including the standard density-based SIMP methods, discrete-variable-based methods, and level-set-based approaches. We highlight that the goal of the numerical example is not to compare the performance and capability of different topology optimization methods, but to demonstrate the versatility and usability of those codes and report an overall user experience. Thus, all codes were run "as is" with default settings.

The design domain of the cantilever beam and the associated boundary conditions are shown in Fig. 6. Unless otherwise specified, a FE mesh of the size $200 \times 100 = 20,000$ is employed. The Young's modulus of the solid material is set as $E_0 = 1$ and the ersatz Young's modulus of the void phase is taken to be $E_{min} = 10^{-9}$. A Poisson's ratio of $\nu = 0.3$ is assigned to both solid and void phases. Notice that, we aim to run those codes using their default settings. The convergence criteria (e.g., maximum iteration and tolerance) adopted by each code are different (according to their default settings) and thus could lead to the different numbers of total optimization iterations used. In order to report the compliance values in a consistent manner (for both optimization histories and optimized designs), we normalize them by the compliance value of an initial design with uniform density distribution (which satisfies volume constraint and with the penalization parameter $p = 1$).

Figure 7 plots the optimized results obtained from the four standard density-based SIMP codes, `top99` (Sigmund 2001), `top88` (Andreassen et al. 2011), `PolyTop` (Talischi et al. 2012b) and `top99neo` (Ferrari and Sigmund 2020). The detailed setups of the FE discretization, material interpolation, filtering, and design updated schemes adopted in each code are summarized as follows. For the FE discretization, the structured quadrilateral meshes are used in `top99`, `top88`, and `top99neo`, while the `PolyTop` employs a polygonal discretization. For the material interpolation scheme, the classical SIMP method (i.e., Eq. (2)) is used in the `top99`, while the others use the modified SIMP approach (i.e., Eq. (3)). We fix the SIMP penalization parameters to be $p = 3$ in these codes. For `PolyTop`, we also present an additional result obtained with the default setup of a continuation of parameter p from 1 to 3 with an interval of 0.5. For the filtering, sensitivity filters are used in `top99` and `top88`, and the density filters are adopted in `top88`, `PolyTop`, and `top99neo`. In addition, an optimized result obtained by using the density filter together with the Heaviside projection (Wang et al. 2011) is also presented for `top99neo`. For all cases, the filter radius r_{min} is set to be $0.06L$. For the design update scheme, the OC method is employed for

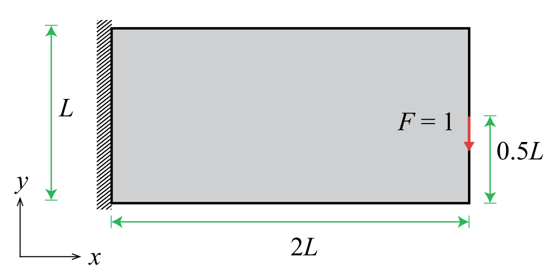
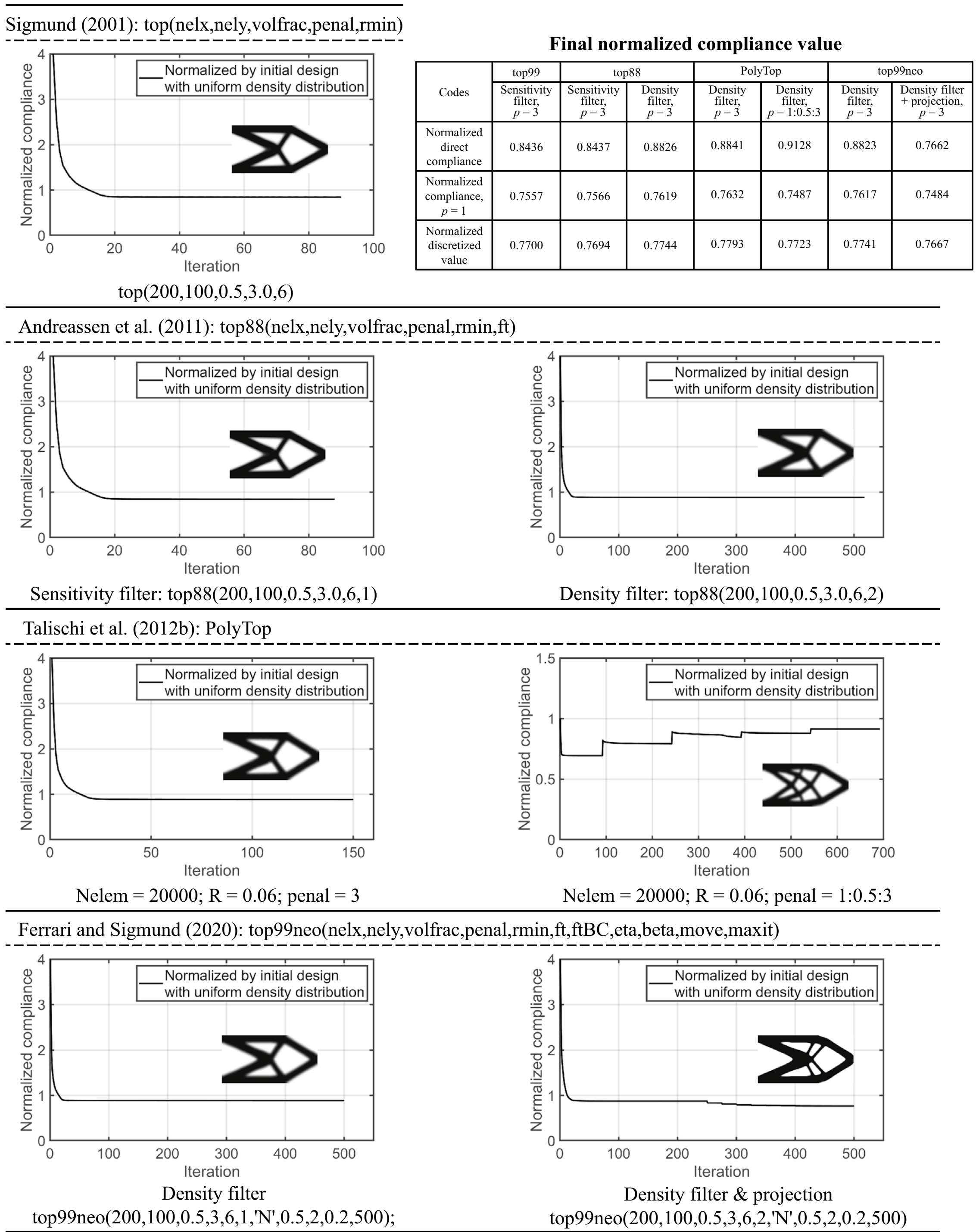


Fig. 6 Design domain of the cantilever beam. (Various codes may have different setups for dimensions, thus we report the normalized compliance values so that the magnitude of L does not influence the result.)



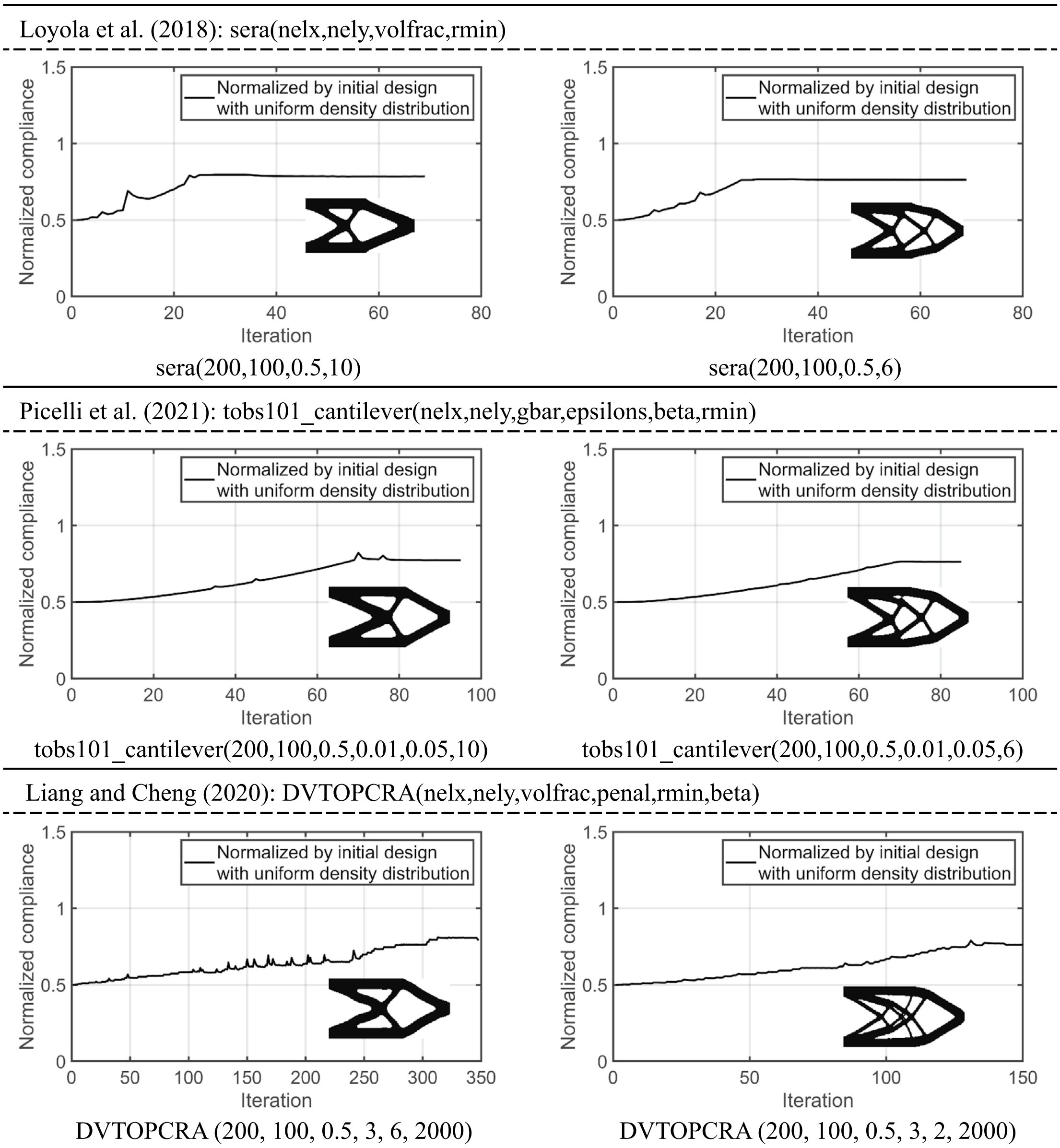
Final normalized compliance value

Codes	top99	top88		PolyTop		top99neo	
	Sensitivity filter, $p = 3$	Sensitivity filter, $p = 3$	Density filter, $p = 3$	Density filter, $p = 3$	Density filter, $p = 1{:}0.5{:}3$	Density filter, $p = 3$	Density filter + projection, $p = 3$
Normalized direct compliance	0.8436	0.8437	0.8826	0.8841	0.9128	0.8823	0.7662
Normalized compliance, $p = 1$	0.7557	0.7566	0.7619	0.7632	0.7487	0.7617	0.7484
Normalized discretized value	0.7700	0.7694	0.7744	0.7793	0.7723	0.7741	0.7667

Fig. 7 Cantilever beam example generated by standard density-based SIMP codes

all the codes. We remark that the main parameters in these SIMP codes are the filter radius r_{min} and the penalization parameter p, which are physical parameters. In practice, the filter radius r_{min} should be determined based on design requirements such as member length scales or design complexity. The penalization parameter p is suggested to either take $p = 3$ or follow a continuation strategy (e.g., gradually increases p from 1 to 3).

From the final results presented, we observe that, under the same parameter setup, similar optimized geometries can be obtained from different SIMP codes on various (e.g., quadrilateral and polygonal) discretizations, which indicate



Final normalized compliance value

Codes	sera		tobs101_cantilever		DVTOPCRA	
	$r_{min} = 10$	$r_{min} = 6$	$r_{min} = 10$	$r_{min} = 6$	$r_{min} = 6$	$r_{min} = 2$
Normalized direct compliance	0.7846	0.7621	0.7741	0.7644	0.7953	0.7614

Fig. 8 Cantilever beam example generated by discrete variables codes

their consistency. With the continuation in the penalization parameter p, a different design is obtained with `PolyTop`, demonstrating the influence of penalization parameter p on the optimization results. For this case, each increase of compliance value in the convergence history corresponds to an increase in the p value. Because of the continuous nature of the design variables and the filtering, most optimized designs (except for the one by adopting Heaviside projection) obtained by the standard SIMP codes contain grayscale elements, which have an influence on their compliance values. Thus, Fig. 7 also contains a table that reports two additional compliance values for each obtained optimized design according to the two realizations of gray-scale designs. In the first realization, we evaluate the compliance

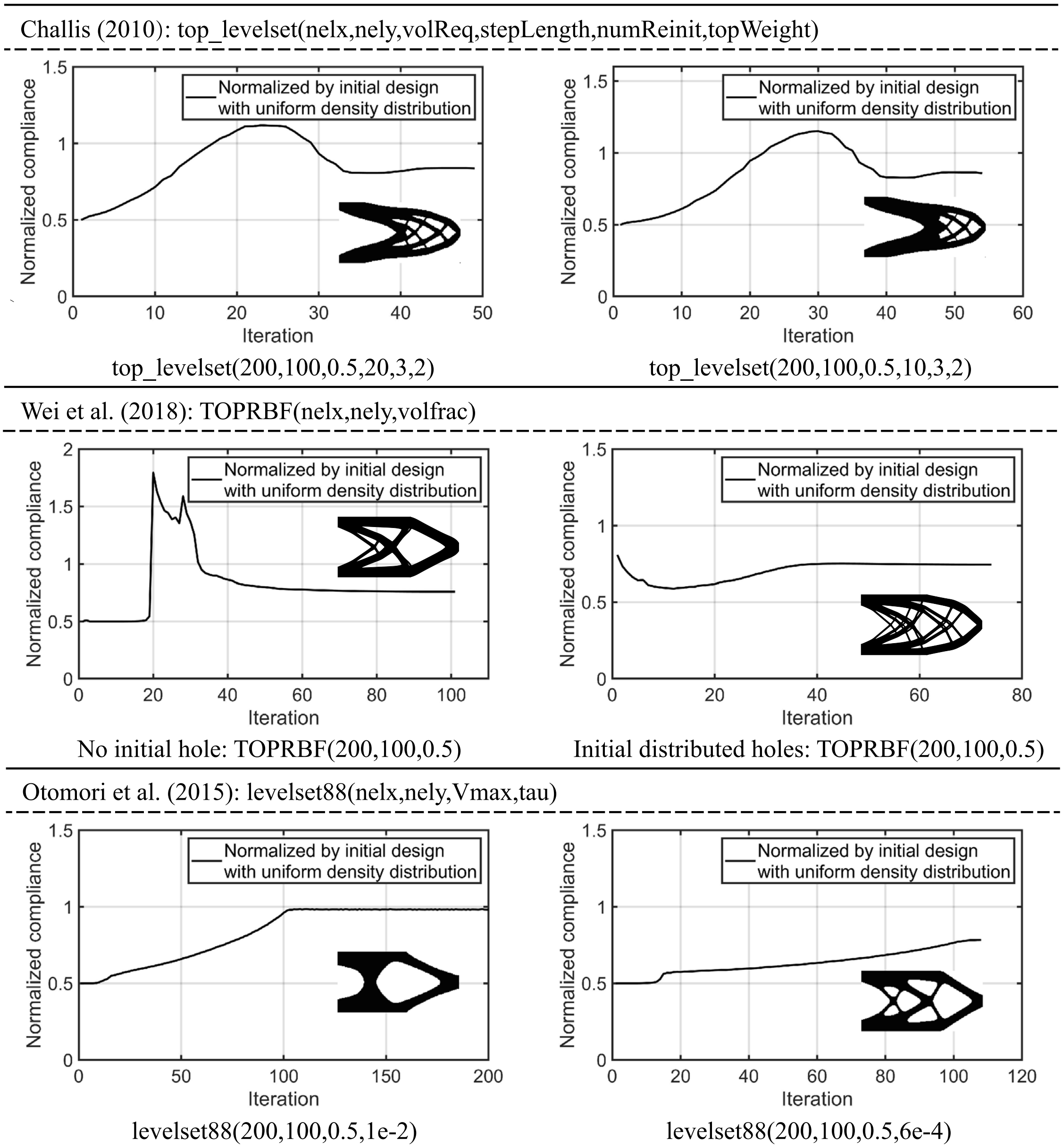
Journal : **Large 158**	Article No : **3050**	Pages : **54**	MS Code : **3050**	Dispatch : **28-9-2021**

of the optimized designs with gray-scale elements by setting $p = 1$ in the SIMP material interpolation. For 2D problems, this transforms the designs into variable-thickness sheets where the density represents the out-of-plane thickness. In the second realization, we make use of the volume-conserving post-processing technique suggested in Sigmund and Maute (2013) to map the gray-scale designs into binary ones and then evaluate the compliance values of those post-processed binary designs. We remark that, if the readers were to compare the performance of the optimized designs between the SIMP method and other methods, they are suggested to use the compliance values associated with the post-processed binary designs (Sigmund and Maute 2013).

The design results and convergence histories for the three codes employing the discrete variable-based methods are shown in Fig. 8. The first code, `sera` (Loyola et al. 2018), adopts the bi-directional SERA method. The other two codes, `tobs101` (Picelli et al. 2021) and `DVTOPCRA` (Liang and Cheng 2020), solve the discrete variable topology optimization via integer programming. The code `tobs101` employs a branch-and-bound solver, whereas the code `DVTOPCRA` utilizes a canonical relaxation algorithm. It is worth noting that all the three codes are developed based on the standard SIMP codes. The first code `sera` is built upon `top99`, where the main difference lies in the material update subroutine. The latter two codes `tobs101` and `DVTOPCRA` use the same convention in their FE and sensitivity analyses as the `top88`. All the three codes employ the sensitivity filter to alleviate the mesh-dependent issues. For the first two codes (`sera` and `tobs101`), we observe that setting the same $r_{min} = 0.06L = 6$ typically produces optimized designs with different topologies from the ones obtained by the standard SIMP codes. We think this is a consequence of the nonconvexity and the discrete nature of design variables. By adjusting the filter radius to $r_{min} = 0.1L = 10$, designs with similar topologies to the ones generated by the SIMP codes can be obtained. For the code `DVTOPCRA`, besides the result generated using $r_{min} = 0.06L = 6$ as that used in the standard SIMP codes resulting in a similar topology, we also report the result obtained using the default parameter setups (i.e., $r_{min} = 0.02L = 2$) for the given cantilever beam example. In addition to the filter radius, the codes `tobs101` and `DVTOPCRA` also contain several user-specified parameters related to the respective integer programming solvers and provide guidance on how to choose them. For the code `tobs101`, these parameters include the constraint relaxation parameter `epsilons`, which restricts the decreasing proportion of volume in each step, and the truncation error constraint `beta`, which restricts the number of flips on design variables. It is demonstrated in Picelli et al. (2021) that the choice of these two parameters should satisfy the relationship of `epsilons` $\leq$ `beta` for problems with a single volume constraint. In our numerical experimentation, default values of these two parameters (i.e., `epsilons` $= 0.01$ and `beta` $= 0.05$) are employed. For the code `DVTOPCRA`, there are two major tuning parameters related to the canonical relaxation algorithm: the perturbed parameter `beta` and the initial dual variable λ, whose effects and guides are provided by the authors (Liang and Cheng 2020). Accordingly, default values of `beta` $= 2000$ and $\lambda = 10^{-3}$ are used in our example.

The results obtained using the three level-set topology optimization approaches are demonstrated in Fig. 9. The first code is for a discrete level-set topology optimization (Challis 2010). The tuning parameters in this code include `stepLength` to specify the time interval for evolving the level-set function, `numReinit` to determine the reinitialization frequency, and `topWeight` to assign the weight of topological derivative in the evolution equation. Suggestions on the suitable value ranges for these parameters and potential effects are provided by the authors. According to the recommended range of `stepLength`, which is an integer value between min(`nelx`, `nely`)/10 and max(`nelx`, `nely`)/5, Fig. 9 shows the results obtained under two values of `stepLength` (i.e., `stepLength` $= 20$ and 10). Although the code generates similar designs under these two values of `stepLength`, it is noticed that the topologies of these two designs are quite different from those obtained by other codes. Thus, the usability of this code remains to be further verified by users. The second code `TOPRBF` (Wei et al. 2018) implements a parameterized level-set method using the radial basis functions. The parameters of this code are related to the Lagrange multiplier computation as well as the time step interval in the evolution scheme. Two types of initial guesses, one without initial holes and the other with distributed initial holes, are investigated. The results demonstrate that this code is capable of creating new holes inside the design domain during optimization. The last one, `levelset88` (Otomori et al. 2015), is a MATLAB code which implements the level-set topology optimization using a reaction-diffusion equation approach. The tuning parameter in this code is the regularization parameter `tau` in the reaction-diffusion equation. The influence of different values of `tau` is investigated by the authors in the original article. It is suggested that a larger `tau` results in a design with less complexity in its geometry and vice versa. In this case, the complexity in the optimized topology can be controlled via adjusting this regularization parameter `tau`. This suggestion is also verified by the results shown in Fig. 9 obtained by two different values of `tau`.

Journal : Large 158	Article No : 3050	Pages : 54	MS Code : 3050	Dispatch : 28-9-2021



Final normalized compliance value

Codes	top_levelset		TOPRBF		levelset88	
	stepLength = 20	stepLength = 10	No initial hole	Initial distributed holes	tau = 1e-2	tau = 6e-4
Normalized direct compliance	0.8358	0.8587	0.7578	0.7452	0.9816	0.7828

Fig. 9 Cantilever beam example generated by level-set method codes

8 Conclusions and perspectives

The field of structural and multidisciplinary optimization (SMO) has made great progress over the past decades. Accompanying the development of various SMO methods, educational articles have become an increasingly popular genre and have made considerable contributions to the field. This review paper aims to provide a comprehensive survey of educational and other types of papers, with a particular focus on codes that provide a complete immersive experience. To provide a clear overview we grouped contributions in categories based on problems and methods. Educational codes are assessed on their usability, efficiency, compactness, and readability. A comparative study is given on select codes to shed light on user experiences, results consistency, and code robustness. This section can be particularly helpful for students and newcomers of the field. We also provided insights of codes as building blocks that can be used by researchers to implement their own research projects.

In addition, we would like to offer some general observations and forward-looking recommendations:

1. As shown in Fig. 1, the quantity of educational papers has continued to accelerate in recent years. While the trend is overwhelmingly positive, we also observed some early signs of potential oversupply of educational content. Given the more competitive landscape, authors should strive for a more clear emphasis on educational impact. Educational values are typically reflected by one or more of the following components:

    (a) Introducing a noteworthy method to students. Here the focus is on exposing a proven major approach to students and newcomers to the SMO field.
    (b) For educational purposes, the article should ideally have a self-contained theory and formulation content. The basic version of the code should emphasize readability and easy understanding of the numerical and implementation details. More efficient version(s) using advanced programming techniques can be included as appendices or ESM (electronic supplemental material). The paper should also provide sufficient insights on the strengths and weaknesses of the underlying method.
    (c) Elucidating solutions for important engineering and science problems that are considered complex and challenging. Such problems include but are not limited to: (i) more challenging performance constraints such as stress, buckling, etc.; (ii) more complex structural analysis types including large deformation, material nonlinearity, history- or rate-dependent materials such as viscoelasticity; (iii) more complex or multiple physics such as thermal, fluid flow, acoustics, electromagnetics, etc. Analysis and optimization codes for these problems are less available compared to structural solutions. Hence, self-contained codes providing hands-on experience could offer significant educational value for students and fellow researchers. They could also help to accelerate software advancements and industrial applications.
    (d) Another type of educational paper could aim at exposing students and researchers to new programming platforms, languages, techniques, and toolboxes with the purposes of (i) easy creation of solutions; (ii) increasing computational efficiency; (iii) building and sustaining open-source communities.
    (e) No-code-based educational papers are also welcome if they help dissecting complex theories and formulations into highly teachable forms.

2. Sharing source codes as part of a research paper has increasingly become a common practice for many fields such as statistics and computer science. Our field has also been trending in this direction, especially since the SMO journal made replication of results a mandatory section. Authors are more aware of the positive effect on the impact of their work from code and data sharing. As ESM becomes widely available for journal publications, it would not be the best approach to branch out code sharing into an educational paper, unless significant educational contents are warranted.
3. Educational codes should be made stable and modular with clearly structured components. Specifically, the codes should be accompanied by: (a) detailed comments of each module; (b) clear specification and guidelines on user parameters (physical parameters such as minimum/maximum length scale, and tuning parameters), with a clear indication if physical parameters are guaranteed in results. Moreover, having to change tuning parameters for problems with different geometry, loading and boundary conditions should be avoided; (c) computing environment settings and dependent platforms and tools. In addition, a step-by-step checklist should be provided to make the user experience seamless.
4. Our experience studying codes with historical evolutionary trees (see Figs. 4 and 5) shows that there are clear benefits when a code is developed from previous code generations that are widely used. Users can jump start their immersive experience quickly due to familiarity of the building blocks and coding structure and style. Also, it helps authors to reduce development effort considerably. This would be a highly recommended approach whenever possible. Even a brand new code following a familiar style would make it much more accessible to a user.
5. Educational codes written in MATLAB should check compatibility with the alternative open-source platform—GNU Octave. In addition, for a plug-and-play experience authors should always provide editable source code.
6. For meaningful performance comparison between results obtained by different methods and/or options,

Journal : Large 158	Article No : 3050	Pages : 54	MS Code : 3050	Dispatch : 28-9-2021

effects of intermediate density should be removed. We recommend two alternative approaches: (a) run a final analysis with $p = 1$ in SIMP or the equivalent for other methods if penalty effects exist; (b) run a final analysis after post-processed design into discrete 0-1 results using code snippet from Sigmund and Maute (2013). Performance comparison shown in Fig. 7 followed the above approaches.

7. For beginners and for classroom teaching of SMO methods, we recommend starting from the basic codes and interactive apps and moving on to the advanced codes that focus on efficiency and/or other problems (e.g., multiphysics). For this purpose, Tables 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 and 13 (with DOI information) can serve as a dictionary for readers to quickly identify a suitable code and corresponding reference.

It is worth noting that educational papers have, in many ways, a game-changing effect on the rapid growth of research content and depth in the SMO field. As the vast majority of research work are carried out by Ph.D. students, the availability of compact codes covering wide-ranging problems helps to shorten students' learning curve tremendously. Moreover, the familiarity of working codes helps to launch students, researchers, and industrial developers into their own research experiments seamlessly. The significant usage and citation data shown in Fig. 1 are clear evidence of the compounding effects and impact of educational contributions. We hope this survey can help researchers, especially newcomers, gaining a quick overview of a large set of available codes and educational content. We also hope that our observations and suggestions can help to further enlarge the impact and influence of high-quality educational contributions going forward.

Acknowledgements The authors would like to devote this paper to Dr. Rafi Hafka. Rafi was exemplary as an educator throughout his legendary research career. It is not a coincidence that most educational papers have appeared in the SMO journal. This publication genre was deliberately cultivated by Haftka, and also his predecessor Rozvany, as Editor-in-Chief of the journal. It is most fitting and gratifying to have this review included in the Special Issue dedicated to our mentor and dear friend Rafi Haftka. The authors would like to acknowledge the following financial supports. C. Wang, Z. Zhao, and X.S. Zhang were supported by the U.S. National Science Foundation (NSF) CAREER Award CMMI-2047692. O. Sigmund was supported by the Villum Foundation through the Villum Investigator Project "InnoTop". The information provided in this paper is the sole opinion of the authors and does not necessarily reflect the view of the sponsoring agencies.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Replication of results All results in this paper are generated by codes and data from source references. Readers are encouraged to download papers and codes of interest from original publications.

References

Aage N, Nobel-Jørgensen M, Andreasen CS, Sigmund O (2013) Interactive topology optimization on hand-held devices. Struct Multidisc Optim 47(1):1–6

Aage N, Andreassen E, Lazarov BS (2015) Topology optimization using PETSc: an easy-to-use, fully parallel, open source topology optimization framework. Struct Multidisc Optim 51(3):565–572

ABAQUS Inc (2021) ABAQUS. https://www.3ds.com/products-services/simulia/products/abaqus/

Allaire G, Pantz O (2006) Structural optimization with FreeFem++. Struct Multidisc Optim 32(3):173–181

Allaire G, Jouve F, Toader AM (2002) A level-set method for shape optimization. CR Math 334(12):1125–1130

Amir O (2015) Revisiting approximate reanalysis in topology optimization: on the advantages of recycled preconditioning in a minimum weight procedure. Struct Multidisc Optim 51(1):41–57

Amir O, Aage N, Lazarov BS (2014) On multigrid-CG for efficient topology optimization. Struct Multidisc Optim 49(5):815–829

Andreasen CS, Elingaard MO, Aage N (2020) Level set topology and shape optimization by density methods using cut elements with length scale control. Struct Multidisc Optim 62(2):685–707

Andreassen E, Andreasen CS (2014) How to determine composite material properties using numerical homogenization. Comput Mater Sci 83:488–495

Andreassen E, Clausen A, Schevenels M, Lazarov BS, Sigmund O (2011) Efficient topology optimization in MATLAB using 88 lines of code. Struct Multidisc Optim 43(1):1–16

Ansys Inc (2021) Ansys. https://www.ansys.com/

Bai J, Zuo W (2020) Hollow structural design in topology optimization via moving morphable component method. Struct Multidisc Optim 61(1):187–205

Balay S, Abhyankar S, Adams M, Brown J, Brune P, Buschelman K, Dalcin L, Dener A, Eijkhout V, Gropp W, et al. (2019) PETSc users manual

Beckers P, Beckers B (2015) A 66 line heat transfer finite element code to highlight the dual approach. Comput Math Appl 70(10):2401–2413

Beirão da Veiga L, Brezzi F, Cangiani A, Manzini G, Marini LD, Russo A (2013) Basic principles of virtual element methods. Math Models Methods Appl Sci 23(01):199–214

Belotti R, Richiedei D, Trevisani A (2021) Multi-domain optimization of the eigen structure of controlled underactuated vibrating systems. Struct Multidisc Optim 63(1):499–514

Bendsøe MP (1989) Optimal shape design as a material distribution problem. Struct Optim 1(4):193–202

Bendsøe MP, Sigmund O (1995) Optimization of structural topology, shape, and material, vol 414. Springer, New York

Bendsøe MP, Sigmund O (1999) Material interpolation schemes in topology optimization. Arch Appl Mech 69(9):635–654

Bendsøe MP, Sigmund O (2013) Topology optimization: theory, methods, and applications. Springer, New York

Biyikli E, To AC (2015) Proportional topology optimization: a new non-sensitivity method for solving stress constrained and

Journal : Large 158	Article No : 3050	Pages : 54	MS Code : 3050	Dispatch : 28-9-2021

minimum compliance problems and its implementation in MATLAB. PLoS ONE 10(12):e0145041

Bletzinger KU (2014) A consistent frame for sensitivity filtering and the vertex assigned morphing of optimal shape. Struct Multidisc Optim 49(6):873–895

Borrvall T, Petersson J (2003) Topology optimization of fluids in stokes flow. Int J Numer Meth Fluids 41(1):77–107

Burger M, Stainko R (2006) Phase-field relaxation of topology optimization with local stress constraints. SIAM J Control Optim 45(4):1447–1466

Challis VJ (2010) A discrete level-set topology optimization code written in Matlab. Struct Multidisc Optim 41(3):453–464

Chandrasekhar A, Suresh K (2021) TOuNN: Topology optimization using neural networks. Struct Multidisc Optim 63(3):1135–1149

Chandrasekhar A, Sridhara S, Suresh K (2021) AuTO: A framework for automatic differentiation in topology optimization. arXiv:210401965

Chen Q, Zhang X, Zhu B (2019) A 213-line topology optimization code for geometrically nonlinear structures. Struct Multidisc Optim 59(5):1863–1879

Chi H, Pereira A, Menezes IFM, Paulino GH (2020) Virtual element method (VEM)-based topology optimization: an integrated framework. Struct Multidisc Optim 62(3):1089–1114

Christiansen RE, Sigmund O (2021) Compact 200 line MATLAB code for inverse design in photonics by topology optimization: tutorial. J Opt Soc Am B 38(2):510–520

Christiansen RE, Sigmund O (2021) Inverse design in photonics by topology optimization: tutorial. J Opt Soc Am B 38(2):496–509

Chung H, Hwang JT, Gray JS, Kim HA (2019) Topology optimization in OpenMDAO. Struct Multidisc Optim 59(4):1385–1400

COMSOL AB (2021) COMSOL. https://www.comsol.com/

Coniglio S, Morlier J, Gogu C, Amargier R (2020) Generalized geometry projection: a unified approach for geometric feature based topology optimization. Arch Comput Methods Eng 27:1573–1610

Csébfalvi A (2017) Robust topology optimization: a new algorithm for volume-constrained expected compliance minimization with probabilistic loading directions using exact analytical objective and gradient. Periodica Polytechnica Civil Eng 61(1):154–163

Da D, Xia L, Li G, Huang X (2018) Evolutionary topology optimization of continuum structures with smooth boundary representation. Struct Multidisc Optim 57(6):2143–2159

Dapogny C, Frey P, Omnes F, Privat Y (2018) Geometrical shape optimization in fluid mechanics using FreeFem++. Struct Multidisc Optim 58(6):2761–2788

Deaton JD, Grandhi RV (2014) A survey of structural and multidisciplinary continuum topology optimization: post 2000. Struct Multidisc Optim 49(1):1–38

Deng H, Vulimiri PS, To AC (2021) An efficient 146-line 3D sensitivity analysis code of stress-based topology optimization written in MATLAB. arXiv:210401210

Dong G, Tang Y, Zhao YF (2019) A 149 line homogenization code for three-dimensional cellular materials written in matlab. J Eng Mater Technol 141(1):011005

Dunning PD (2020) On the co-rotational method for geometrically nonlinear topology optimization. Struct Multidisc Optim 62(5):2357–2374

Dzierzanowski G (2012) On the comparison of material interpolation schemes and optimal composite properties in plane shape optimization. Struct Multidisc Optim 46(5):693–710

Elham A, van Tooren MJ (2021) Discrete adjoint aerodynamic shape optimization using symbolic analysis with OpenFEMflow. Struct Multidisc Optim 63(5):2531–2551

Eschenauer HA, Kobelev VV, Schumacher A (1994) Bubble method for topology and shape optimization of structures. Struct Optim 8(1):42–51

ESI Group (2021) Scilab. https://www.scilab.org/

Evgrafov A (2015) On Chebyshev's method for topology optimization of Stokes flows. Struct Multidisc Optim 51(4):801–811

Fairclough H, He L, Pritchard T, Gilbert M (2021) LayOpt: an educational web-app for truss layout optimization. Struct Multidisc Optim

Fernández E, Collet M, Alarcón P, Bauduin S, Duysinx P (2019) An aggregation strategy of maximum size constraints in density-based topology optimization. Struct Multidisc Optim 60(5):2113–2130

Ferrari F, Sigmund O (2020) A new generation 99 line Matlab code for compliance topology optimization and its extension to 3D. Struct Multidisc Optim 62(4):2211–2228

Ferrari F, Sigmund O, Guest JK (2021) Topology optimization with linearized buckling criteria in 250 lines of Matlab. Struct Multidisc Optim 63(6):3045–3066

Filomeno Coelho R, Tysmans T, Verwimp E (2014) Form finding & structural optimization: a project-based course for graduate students in civil and architectural engineering. Struct Multidisc Optim 49(6):1037–1046

Fu YF, Rolfe B, Chiu LNS, Wang Y, Huang X, Ghabraie K (2020) SEMDOT: smooth-edged material distribution for optimizing topology algorithm. Adv Eng Softw 150:102921

Gangl P, Sturm K, Neunteufel M, Schöberl J (2021) Fully and semi-automated shape differentiation in NGSolve. Struct Multidisc Optim 63(3):1579–1607

Gao J, Gao L, Luo Z, Li P (2019a) Isogeometric topology optimization for continuum structures using density distribution function. Int J Numer Meth Eng 119(10):991–1017

Gao J, Luo Z, Xia L, Gao L (2019) Concurrent topology optimization of multiscale composite structures in Matlab. Struct Multidisc Optim 60(6):2621–2651

Gao J, Wang L, Luo Z, Gao L (2021) IgaTop: an implementation of topology optimization for structures using IGA in MATLAB. Struct Multidisc Optim

Ghantasala A, Asl RN, Geiser A, Brodie A, Papoutsis E, Bletzinger KU (2021) Realization of a framework for simulation-based large-scale shape optimization using vertex morphing. J Optim Theory Appl 189(1):164–189

Giraldo-Londoño O, Paulino GH (2021a) PolyDyna: a Matlab implementation for topology optimization of structures subjected to dynamic loads. Struct Multidisc Optim 64:957–990

Giraldo-Londoño O, Paulino GH (2021b) PolyStress: a Matlab implementation for local stress-constrained topology optimization using the augmented Lagrangian method. Struct Multidisc Optim 63(4):2065–2097

Gray JS, Hwang JT, Martins JRRA, Moore KT, Naylor BA (2019) OpenMDAO: an open-source framework for multidisciplinary design, analysis, and optimization. Struct Multidisc Optim 59(4):1075–1104

Guo X, Zhang W, Zhong W (2014) Doing topology optimization explicitly and geometrically-a new moving morphable components based framework. J Appl Mech 81(8):1

Haftka R, Jenkins D (1998) Classroom project in analytical and experimental optimization. Struct Optim 15(1):63–67

Han H, Guo Y, Chen S, Liu Z (2021a) Topological constraints in 2D structural topology optimization. Struct Multidisc Optim 63(1):39–58

Han Y, Xu B, Liu Y (2021b) An efficient 137-line MATLAB code for geometrically nonlinear topology optimization using bi-directional evolutionary structural optimization method. Struct Multidisc Optim 63(5):2571–2588

He L, Gilbert M, Song X (2019) A Python script for adaptive layout optimization of trusses. Struct Multidisc Optim 60(2):835–847

Hecht F (2012) New development in FreeFem++. J Numer Math 20(3–4):251–266. https://doc.freefem.org/documentation/index.html

Homayouni-Amlashi A, Schlinquer T, Mohand-Ousaid A, Rakotondrabe M (2021) 2D topology optimization MATLAB codes for piezoelectric actuators and energy harvesters. Struct Multidisc Optim 63(2):983–1014

Huang X (2021) On smooth or 0/1 designs of the fixed-mesh element-based topology optimization. Adv Eng Softw 151:102942

Huang X, Xie YM (2010) A further review of ESO type methods for topology optimization. Struct Multidisc Optim 41(5):671–683

Huang ZL, Jiang C, Zhang Z, Zhang W, Yang TG (2019) Evidence-theory-based reliability design optimization with parametric correlations. Struct Multidisc Optim 60(2):565–580

Ibhadode O, Zhang Z, Bonakdar A, Toyserkani E (2021) IbIPP for topology optimization-an image-based initialization and post-processing code written in MATLAB. SoftwareX 14:100701

Jasa JP, Hwang JT, Martins JRRA (2018) Open-source coupled aerostructural optimization using Python. Struct Multidisc Optim 57(4):1815–1827

Jensen KE (2018) Topology optimization of stokes flow on dynamic meshes using simple optimizers. Comput Fluids 174:66–77

Kanno Y (2020) On three concepts in robust design optimization: absolute robustness, relative robustness, and less variance. Struct Multidisc Optim 62:979–1000

Keshavarzzadeh V, Kirby RM, Narayan A (2019) Parametric topology optimization with multiresolution finite element models. Int J Numer Meth Eng 119(7):567–589

Kharmanda G, Olhoff N, Mohamed A, Lemaire M (2004) Reliability-based topology optimization. Struct Multidisc Optim 26(5):295–307

Kim C, Jung M, Yamada T, Nishiwaki S, Yoo J (2020) FreeFEM++ code for reaction-diffusion equation-based topology optimization: for high-resolution boundary representation using adaptive mesh refinement. Struct Multidisc Optim 62(1):439–455

Klarbring A (2015) Design optimization based on state problem functionals. Struct Multidisc Optim 52(2):417–425

Krogh C, Bak BL, Lindgaard E, Olesen AM, Hermansen SM, Broberg PH, Kepler JA, Lund E, Jakobsen J (2021) A simple MATLAB draping code for fiber-reinforced composites with application to optimization of manufacturing process parameters. Struct Multidisc Optim

Kumar T, Suresh K (2021) Direct Lagrange multiplier updates in topology optimization revisited. Struct Multidisc Optim 63(3):1563–1578

Lagaros ND, Vasileiou N, Kazakis G (2019) A C# code for solving 3D topology optimization problems using SAP2000. Optim Eng 20(1):1–35

Langelaar M (2017) An additive manufacturing filter for topology optimization of print-ready designs. Struct Multidisc Optim 55(3):871–883

Langtangen HP, Logg A (2017) Solving PDEs in python: the FEniCS tutorial I. Springer, New York

Laurain A (2018) A level set-based structural optimization code using FEniCS. Struct Multidisc Optim 58(3):1311–1334

Lelièvre N, Beaurepaire P, Mattrand C, Gayton N, Otsmane A (2016) On the consideration of uncertainty in design: optimization—reliability—robustness. Struct Multidisc Optim 54(6):1423–1437

Li HS, Cao ZJ (2016) Matlab codes of Subset Simulation for reliability analysis and structural optimization. Struct Multidisc Optim 54(2):391–410

Liang Y, Cheng G (2020) Further elaborations on topology optimization via sequential integer programming and Canonical relaxation algorithm and 128-line MATLAB code. Struct Multidisc Optim 61(1):411–431

Lin H, Xu A, Misra A, Zhao R (2020) An ANSYS APDL code for topology optimization of structures with multi-constraints using the BESO method with dynamic evolution rate (DER-BESO). Struct Multidisc Optim 62(4):2229–2254

Liu K, Tovar A (2014) An efficient 3D topology optimization code written in Matlab. Struct Multidisc Optim 50(6):1175–1196

Liu Z, Korvink JG, Huang R (2005) Structure topology optimization: fully coupled level set method via FEMLAB. Struct Multidisc Optim 29(6):407–417

Liu H, Zong H, Tian Y, Ma Q, Wang MY (2019) A novel subdomain level set method for structural topology optimization and its application in graded cellular structure design. Struct Multidisc Optim 60(6):2221–2247

Loyola RA, Querin OM, Jiménez AG, Gordoa CA (2018) A sequential element rejection and admission (SERA) topology optimization code written in Matlab. Struct Multidisc Optim 58(3):1297–1310

Martin A, Deierlein GG (2020) Structural topology optimization of tall buildings for dynamic seismic excitation using modal decomposition. Eng Struct 216:110717

Nguyen TT, Bærentzen JA, Sigmund O, Aage N (2020) Efficient hybrid topology and shape optimization combining implicit and explicit design representations. Struct Multidisc Optim 62(3):1061–1069

Nie Z, Lin T, Jiang H, Kara LB (2021) TopologyGAN: topology optimization using generative adversarial networks based on physical fields over the initial domain. J Mech Des 143(3):031715

Ning A (2021) Using blade element momentum methods with gradient-based design optimization. Struct Multidisc Optim

Nobel-Jørgensen M, Aage N, Nyman Christiansen A, Igarashi T, Andreas Bærentzen J, Sigmund O (2015) 3D interactive topology optimization on hand-held devices. Struct Multidisc Optim 51(6):1385–1391

Nobel-Jørgensen M, Malmgren-Hansen D, Bærentzen JA, Sigmund O, Aage N (2016) Improving topology optimization intuition through games. Struct Multidisc Optim 54(4):775–781

Olesen LH, Okkels F, Bruus H (2006) A high-level programming-language implementation of topology optimization applied to steady-state Navier–Stokes flow. Int J Numer Meth Eng 65(7):975–1001

Osher S, Fedkiw R (2006) Level set methods and dynamic implicit surfaces, vol 153. Springer, New York

Osher S, Sethian JA (1988) Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations. J Comput Phys 79(1):12–49

Otomori M, Yamada T, Izui K, Nishiwaki S (2015) Matlab code for a level set-based topology optimization method using a reaction diffusion equation. Struct Multidisc Optim 51(5):1159–1172

Overvelde JT (2012) The moving node approach in topology optimization. Delft University of Technology

Paganini A, Wechsung F (2021) Fireshape: a shape optimization toolbox for Firedrake. Struct Multidisc Optim 63(5):2553–2569

Pereira A, Talischi C, Paulino GH, Menezes IFM, Carvalho MS (2016) Fluid flow topology optimization in PolyTop: stability and computational implementation. Struct Multidisc Optim 54:1345–1364. https://doi.org/10.1007/s00158-014-1182-z

Picelli R, Sivapuram R, Xie YM (2021) A 101-line MATLAB code for topology optimization using binary variables and integer programming. Struct Multidisc Optim 63(2):935–954

Rathgeber F, Ham DA, Mitchell L, Lange M, Luporini F, McRae AT, Bercea GT, Markall GR, Kelly PH (2016) Firedrake: automating the finite element method by composing abstractions. ACM Trans Math Softw 43(3):1–27

Rozvany GIN (2009) A critical review of established methods of structural topology optimization. Struct Multidisc Optim 37(3):217–237

Journal : Large 158 | Article No : 3050 | Pages : 54 | MS Code : 3050 | Dispatch : 28-9-2021

Sanders ED, Pereira A, Aguiló MA, Paulino GH (2018) PolyMat: an efficient Matlab code for multi-material topology optimization. Struct Multidisc Optim 58(6):2727–2759

Sangree R, Carstensen JV, Gaynor AT, Zhu M, Guest JK (2015) Topology optimization as a teaching tool for undergraduate education in structural engineering. Struct Congr 2015:2632–2642

Saxena A (2011) Topology design with negative masks using gradient search. Struct Multidisc Optim 44(5):629–649

Schmidt S, Schulz V (2011) A 2589 line topology optimization code written for the graphics card. Comput Vis Sci 14(6):249–256

Schöberl J (2014) C++ 11 implementation of finite elements in NGSolve. Vienna University of Technology, Institute for Analysis and Scientific Computing, Vienna, p 30

Sethian JA (1999) Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science, vol 3. Cambridge University Press, Cambridge

Shaqfa M, Beyer K (2021) Pareto-like sequential sampling heuristic for global optimisation. Soft Computing

Sigmund O (2001) A 99 line topology optimization code written in Matlab. Struct Multidisc Optim 21(2):120–127

Sigmund O (2011) On the usefulness of non-gradient approaches in topology optimization. Struct Multidisc Optim 43(5):589–596

Sigmund O, Maute K (2013) Topology optimization approaches. Struct Multidisc Optim 48(6):1031–1055

Smith H, Norato JA (2020) A MATLAB code for topology optimization using the geometry projection method. Struct Multidisc Optim 62(3):1579–1594

Sokół T (2011) A 99 line code for discretized Michell truss optimization written in mathematica. Struct Multidisc Optim 43(2):181–190

Sokolowski J, Zochowski A (1999) On the topological derivative in shape optimization. SIAM J Control Optim 37(4):1251–1272

Souza B, Yamabe P, Sá L, Ranjbarzadeh S, Picelli R, Silva E (2021) Topology optimization of fluid flow by using integer linear programming. Struct Multidisc Optim

Stolpe M (2010) On some fundamental properties of structural topology optimization problems. Struct Multidisc Optim 41(5):661–670

Stolpe M (2016) Truss optimization with discrete design variables: a critical review. Struct Multidisc Optim 53(2):349–374

Stolpe M (2019) Fail-safe truss topology optimization. Struct Multidisc Optim 60(4):1605–1618

Suresh K (2010) A 199-line Matlab code for Pareto-optimal tracing in topology optimization. Struct Multidisc Optim 42(5):665–679

Svanberg K (1987) The method of moving asymptotes-a new method for structural optimization. Int J Numer Meth Eng 24(2):359–373

Talischi C, Paulino GH, Pereira A, Menezes IFM (2012a) PolyMesher: a general-purpose mesh generator for polygonal elements written in Matlab. Struct Multidisc Optim 45(3):309–328

Talischi C, Paulino GH, Pereira A, Menezes IFM (2012b) PolyTop: a Matlab implementation of a general topology optimization framework using unstructured polygonal finite element meshes. Struct Multidisc Optim 45(3):329–357

Tauzowski P, Blachowski B, Lógó J (2019) Functor-oriented topology optimization of elasto-plastic structures. Adv Eng Softw 135:102690

Tavakoli R (2014) Multimaterial topology optimization by volume constrained Allen–Cahn system and regularized projected steepest descent method. Comput Methods Appl Mech Eng 276:534–565

Tavakoli R, Mohseni SM (2014) Alternating active-phase algorithm for multimaterial topology optimization problems: a 115-line MATLAB implementation. Struct Multidisc Optim 49(4):621–642

Tcherniak D, Sigmund O (2001) A web-based topology optimization program. Struct Multidisc Optim 22(3):179–187

Tyburec M, Zeman J, Doškář M, Kružík M, Lepš M (2021) Modular-topology optimization with wang tilings: an application to truss structures. Struct Multidisc Optim 63(3):1099–1117

van Dijk NP, Maute K, Langelaar M, van Keulen F (2013) Level-set methods for structural topology optimization: a review. Struct Multidisc Optim 48(3):437–472

Wallin M, Ristinmaa M, Askfelt H (2012) Optimal topologies derived from a phase-field method. Struct Multidisc Optim 45(2):171–183

Wang ZP, Kumar D (2017) On the numerical implementation of continuous adjoint sensitivity for transient heat conduction problems using an isogeometric approach. Struct Multidisc Optim 56(2):487–500

Wang MY, Zhou S (2004) Phase field: a variational method for structural topology optimization. CMES 6(6):547

Wang MY, Wang X, Guo D (2003) A level set method for structural topology optimization. Comput Methods Appl Mech Eng 192(1–2):227–246

Wang F, Lazarov BS, Sigmund O (2011) On projection methods, convergence and robust formulations in topology optimization. Struct Multidisc Optim 43(6):767–784

Wei P, Li Z, Li X, Wang MY (2018) An 88-line MATLAB code for the parameterized level set method based topology optimization using radial basis functions. Struct Multidisc Optim 58(2):831–849

Wein F, Dunning PD, Norato JA (2020) A review on feature-mapping methods for structural optimization. Struct Multidisc Optim 62:1597–1638

Wolfram (2021) Mathmatica. https://www.wolfram.com/mathematica/

Wu J, Sigmund O, Groen JP (2021) Topology optimization of multiscale structures: a review. Struct Multidisc Optim 63:1455–1480

Xia L, Breitkopf P (2015) Design of materials using topology optimization and energy-based homogenization approach in Matlab. Struct Multidisc Optim 52(6):1229–1241

Xia L, Xia Q, Huang X, Xie YM (2018) Bi-directional evolutionary structural optimization on advanced structures and materials: a comprehensive review. Arch Comput Methods Eng 25(2):437–478

Xie YM, Steven GP (1993) A simple evolutionary procedure for structural optimization. Comput Struct 49(5):885–896

Xing J, Xu P, Yao S, Zhao H, Zhao Z, Wang Z (2021) A novel weighted graph representation-based method for structural topology optimization. Adv Eng Softw 153:102977

Yaghmaei M, Ghoddosian A, Khatibi MM (2020) A filter-based level set topology optimization method using a 62-line matlab code. Struct Multidisc Optim 62(2):1001–1018

Yago D, Cante J, Lloberas-Valls O, Oliver J (2021) Topology optimization using the unsmooth variational topology optimization (UNVARTOP) method: an educational implementation in MATLAB. Struct Multidisc Optim 63(2):955–981

Yamada T, Izui K, Nishiwaki S, Takezawa A (2010) A topology optimization method based on the level set method incorporating a fictitious interface energy. Comput Methods Appl Mech Eng 199(45–48):2876–2891

Zegard T, Paulino GH (2014) GRAND - Ground structure based topology optimization for arbitrary 2D domains using MATLAB. Struct Multidisc Optim 50(5):861–882

Zegard T, Paulino GH (2015) GRAND3 - Ground structure based topology optimization for arbitrary 3D domains using MATLAB. Struct Multidisc Optim 52(6):1161–1184

Zegard T, Paulino GH (2016) Bridging topology optimization and additive manufacturing. Struct Multidisc Optim 53(1):175–192

Zeng Z, Ma F (2020) An efficient gradient projection method for structural topology optimization. Adv Eng Softw 149:102863

Zhang S, Norato JA, Gain AL, Lyu N (2016a) A geometry projection method for the topology optimization of plate structures. Struct Multidisc Optim 54:1173–1190

Zhang W, Yuan J, Zhang J, Guo X (2016b) A new topology optimization approach based on Moving Morphable Components (MMC) and the ersatz material model. Struct Multidiscip Optim 53(6):1243–1260

Zhang X, Ramos AS, Paulino GH (2017) Material nonlinear topology optimization using the ground structure method with a discrete filtering scheme. Struct Multidiscip Optim 55(6):2045–2072

Zhang XS, Paulino GH, Ramos AS (2018) Multimaterial topology optimization with multiple volume constraints: combining the ZPR update with a ground-structure algorithm to select a single material per overlapping set. Int J Numer Meth Eng 114(10):1053–1073

Zhao Y, Hoang VN, Jang GW, Zuo W (2021) Hollow structural topology optimization to improve manufacturability using three-dimensional moving morphable bars. Adv Eng Softw 152:102955

Zhang ZD, Ibhadode O, Bonakdar A, Toyserkani E (2021) TopADD: a 2D/3D integrated topology optimization parallel-computing framework for arbitrary design domains. Struct Multidisc Optim

Zhou M, Rozvany G (1991) The COC algorithm, Part II: topological, geometrical and generalized shape optimization. Comput Methods Appl Mech Eng 89(1–3):309–336

Zhou M, Sigmund O (2017) On fully stressed design and p-norm measures in structural optimization. Struct Multidisc Optim 56(3):731–736

Zhou M, Sigmund O (2021) Complementary lecture notes for teaching the 99/88-line topology optimization codes. Struct Multidisc Optim. https://doi.org/10.1007/s00158-021-03004-z

Zhou S, Cadman J, Chen Y, Li W, Xie YM, Huang X, Appleyard R, Sun G, Li Q (2012) Design and fabrication of biphasic cellular materials with transport properties—a modified bidirectional evolutionary structural optimization procedure and MATLAB program. Int J Heat Mass Transf 55(25–26):8149–8162

Zhou M, Lian H, Sigmund O, Aage N (2018) Shape morphing and topology optimization of fluid channels by explicit boundary tracking. Int J Numer Meth Fluids 88(6):296–313

Zhu B, Zhang X, Li H, Liang J, Wang R, Li H, Nishiwaki S (2021) An 89-line code for geometrically nonlinear topology optimization written in FreeFEM. Struct Multidisc Optim 63(2):1015–1027

Zuo ZH, Xie YM (2015) A simple and compact Python code for complex 3D topology optimization. Adv Eng Softw 85:1–11