Article:

Victoria, M, Querin, OM, Díaz, C et al. (1 more author) (2016) liteITD a MATLAB Graphical User Interface (GUI) program for topology design of continuum structures. Advances in Engineering Software, 100. pp. 126-147. ISSN 0965-9978

https://doi.org/10.1016/j.advengsoft.2016.07.012

liteITD a MATLAB Graphical User Interface (GUI) program for topology design of continuum structures

Mariano Victoria[1], Osvaldo M. Querin[2], Concepción Díaz[1], and Pascual Martí[1]

[1]Department of Structures and Construction. Technical University of Cartagena

Campus Muralla del Mar. C/Doctor Fleming, s/n. 30202 Cartagena, Spain

E-mail: mariano.victoria@upct.es; conchi.diaz@upct.es; pascual.marti@upct.es

Web: http://www.upct.es/~deyc

[2]School of Mechanical Engineering. University of Leeds

Leeds LS2 9JT, United Kingdom

E-mail: O.M.Querin@leeds.ac.uk

Web: http://www.mech-eng.leeds.ac.uk/menomq/

Keywords: MATLAB GUI, liteITD algorithm, isolines, topology design, continuum structures

Abstract:

Over the past few decades, topology optimization has emerged as a powerful and useful tool for the design of structures, also exploiting the ever growing computational speed and power. The design process has also been affected by computers which changed the concept of form into the concept of formation and the emergence of digital design. Topology optimization can modify existing designs, incorporate explicit features

into a design and generate completely new designs. This paper will show how topology optimization can be used as a digital tool. The liteITD (lite version of Isolines Topology Design) software package will be described with the purpose of providing a tool for topology design. The liteITD program solves the topology optimization of two-dimensional continuum structures using von Mises stress isolines under single or multiple loading conditions, with different material properties in tension and compression, and multiple materials. The liteITD program is fully implemented in the MATrix LABoratory (MATLAB) software environment of MathWorks under Windows operating system. GUIDE (Graphical User Interface Development Environment) was used to create a friendly Graphical User Interface (GUI). The usage of this application is directed to students mainly (educational purposes), although also to designers and engineers with experience. The liteITD program can be downloaded and used for free from the website: http://www.upct.es/goe/software/liteITD.php.

1 INTRODUCTION

In the past, the intuition and experience of designers played a key role in structural design. In the literature, there are a lot of methods available to carry out topology optimization, that can be divided in two types: 1) those with a mathematical basis: Homogenization (Bendsøe and Kikuchi, 1988; Bendsøe, 1995); Solid Isotropic Microstructure with Penalization (SIMP), (Bendsøe, 1989; Rozvany et al., 1992; Bendsøe and Sigmund, 2003); Level Set Method (LSM), (Osher and Sethian, 1988); 2) those based on heuristics: Evolutionary Structural Optimization (ESO), (Xie and Steven, 1997); Bidirectional-ESO (BESO), (Querin et al., 1998) and Isolines Topology Design (ITD), (Victoria et al., 2009). The aim of these methods is to support the intuition and the experience of a designer. For a more exhaust review of topology optimization in structural and continuum mechanics, the reader should refer to Rozvany and Lewiński (2014).

The last two decades have seen the emergence of topology optimization as a very powerful and useful tool for the design of structures. One of the reasons for the growth in the development of topology optimization has been the growth in computational power and speed. This also had a remarkable impact on the design process, shifting it from the traditional paper-based concept of design to digital design. This is self evident by the implementation of topology and other forms of optimization in most commercial structural analysis programs, like ANSYS (ANSYS, 2016), FEMtools (Dynamic design solutions, 2016) and MSC Nastran (MSC software, 2015). There has also been topology optimization specific software with either inbuilt or links to external analysis tools, such as OptiStruct (Altair Engineering, 2016) and CATOPO (Creative Engineering Services, 2016). As well as topology optimization tools available over the internet such as TopOpt (TopOpt, 2009).

The ITD method is easily implemented and can also be called as a subroutine from commercial FEA software packages. However, although the designer can understand how the method works, it is very hard to apply it on demand. Hence, it is necessary to have access to various computer programs such as: finite element analysis software, optimization software, and a good programming background. Besides, the reliance on complicated and expensive third-party programs severely limits the potential use of the ITD method. This paper includes with it a friendly GUI implemented in the technical computation language MATLAB (MathWorks, 2016) that will allow the user to test and use the ITD method.

MATLAB has three advantages over other methods or languages:

1. To provide a software development environment that offers high-performance numerical computation and visualisation capacities.
2. To have a broad spectrum of functions and algorithms for wide range of purposes written by experts, in addition to those written by MATLAB user community.
3. To dispose of a flexible workbench for the integration of other software tools and GUI builder to build up a portable and powerful graphical user interface.

The MATLAB software package has become one of the most popular scientific tools for research and teaching applications. The literature on topology optimization of truss-like continua has a considerable number of available educational computer codes for MATLAB and other platforms. Sigmund (2001) presented a 99 line topology optimization code written in MATLAB for compliance minimization of statically loaded structures. Wang et al. (2004) introduced the 199-line program TOPLSM for the mean compliance optimization of structures in 2D, with the classical level set method. Challis (2010) presented a discrete level-set topology optimization code written in MATLAB. The code can be used to minimize the compliance of a statically loaded structure. Simple code modifications to extend the code for different and multiple loading conditions are given. Suresh (2010) developed a 199-line MATLAB code for Pareto-optimal tracing by exploiting the notion of topological sensitivities. Andreassen et al. (2011) introduced an efficient topology optimization in MATLAB using 88 lines of code with improved assembly and filtering strategies. Liu and Tovar (2014) presented an efficient and compact MATLAB code to solve three-dimensional problems. The 169 lines code solves minimum compliance problems.

On the contrary, the number of applications (apps) or GUIs has relatively few publications. Querin (1997) introduced Evolve97. This computer program is the first ESO program in the world which is developed in the environment of Windows. Tcherniak and Sigmund (2001) presented a web-based interface called Topopt. Paulino et al. (2005) introduced a java-based topology optimization program with web access called I-Top. Coelho and Sierakowski (2008) implemented in MATLAB an educational software tool, called PSOLet, to aid the teaching of Particle Swarm Optimization (PSO) concepts. Zuo (2010) developed BESO2D. This is a standalone program for topology optimization for 2D structures using the latest BESO algorithms. Aage et al. (2013) presented the TopOpt app. This is an interactive topology optimization tool that solves the minimum compliance problem in 2D and is available for iOS and Android devices.

In this paper, an educational topology optimization tool (under Windows operating system) called liteITD (lite version of Isolines Topology Design) is presented to aid the teaching of topology optimization concepts to students mainly, although also to designers and engineers with experience. liteITD is a fully interactive topology design GUI and was implemented in MATLAB software package.

The liteITD consists of three internal modules: pre-processor (geometric modelling, material properties, loading conditions, and supports); solution (Finite Element (FE) mesh generator and FE Analysis (FEA) solver); optimizer (topology optimization algorithms and visualization options). This GUI allows the user to solve the topology optimization of two-dimensional continuum structures using isolines under single or multiple loading conditions, with different material properties in tension and compression, and multiple materials. The topology and the shape of the design depend on an iterative algorithm, which continually adds and removes material depending on the shape and distribution of the contour isolines of the required structural behaviour. In the current liteITD program (version 1.0), only the von Mises stress option is available. Several examples are presented to validate and show the usefulness of the program. The results show the effectiveness of liteITD, providing quality solutions with very detailed shapes without the need to interpret the resulting design.

The liteITD program can be downloaded and used for free from the website: http://www.upct.es/goe/software/liteITD.php. If more detailed information is necessary about liteITD or wish to obtain future upgrades, the user can visit the website or contact us via e-mail at: mariano.victoria@upct.es or O.M.Querin@leeds.ac.uk.

The remainder of this paper is organized as follows. In Section 2, a brief theoretical background of ITD method is given. In Section 3, the liteITD program GUI is described in detail. In Section 4, the liteITD is run from GUI to solve a short cantilever loaded at the centre of the free end. Three more examples and conclusions are discussed in Sections 5 and 6, respectively.

2 THEORETICAL BACKGROUND

The use of isolines and isosurfaces in 2D and 3D, respectively, to obtain the optimum design of structures has been applied in several studies (Woon et al. 2003; Cui et al. 2003; Koguchi and Kikuchi 2006). The ITD is an iterative algorithm which redistributes (adds and removes) material inside a design domain until it reaches a desired volume fraction. The redistribution process consists of the following four steps: (1) obtain the design criterion distribution within the design domain; (2) determine the Minimum Criterion Level (MCL), where its intersection with the design criterion distribution produces the new structural boundary, shown for a 2D continuum in Figure 1; (3) remove all regions from the design domain where the criterion distribution is lower than the MCL; (4) this design modification requires the re-evaluation of the remaining structure in order to recalculate the design criterion distribution.

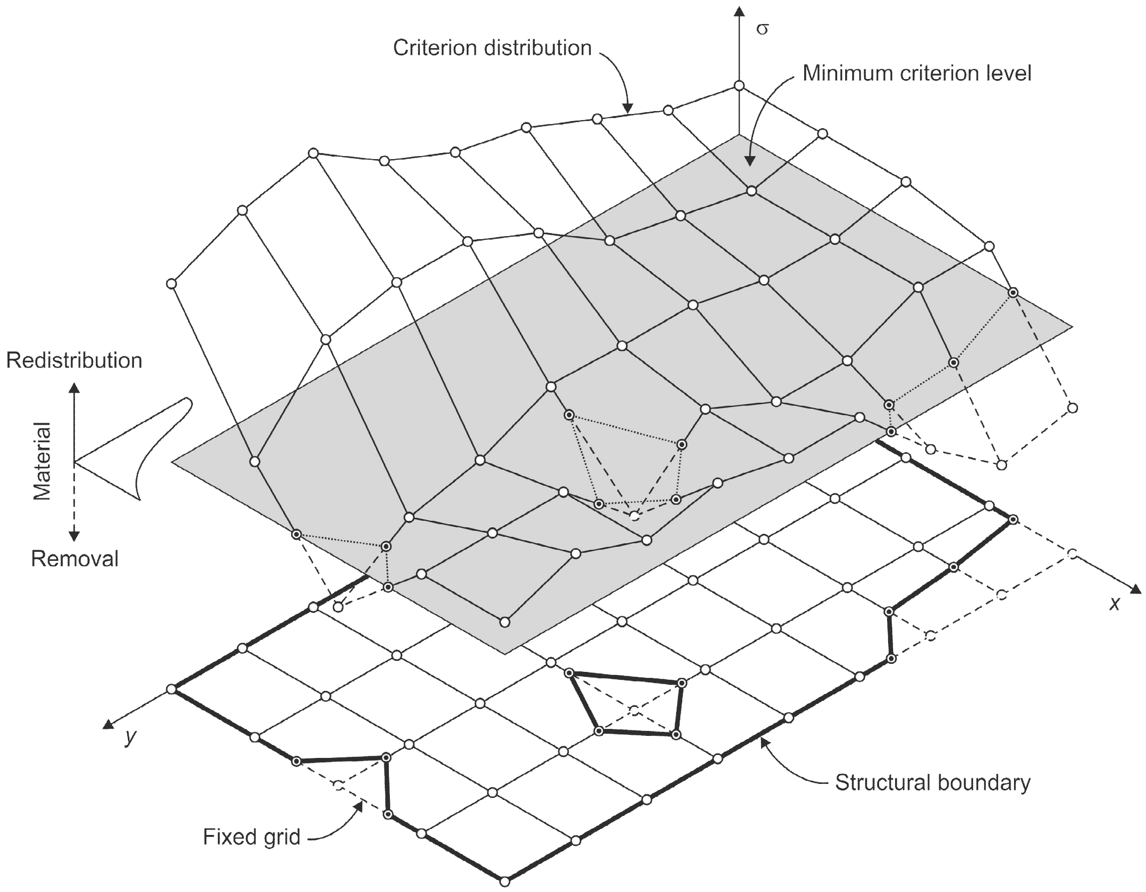

Figure 1. The structural boundary is defined by the intersection of the minimum criterion level (MCL) with the criterion distribution.

The MCL is calculated in each iteration and depends on both the distribution of the design criterion and the volume of the design domain in that iteration, given by Eq. (1):

$$V_i = V_0 + \left(V_f - V_0\right)\frac{i}{n_i} \tag{1}$$

where V_i is the design volume in the ith iteration, V_0 is the initial volume, V_f is the final volume of the optimized structure, and n_i is the total number of iterations.

Once the criterion has been calculated for each element in the design domain, these are arranged in decreasing order of criterion value. An element by element volume summation of the ordered list is carried out until a volume is reached which is as close as possible to the target volume given by Eq. (1), where the error level between the summed and target volume depends on the size of the elements. The criterion value of the next element in the ordered list is then used as the value for the MCL.

2.1 Fixed grid finite element analysis

The Fixed Grid (FG) method was first introduced by Garcia and Steven (1999) as a tool for numerical estimation of 2D elasticity problems, and was later extended to 3D structures by Garcia et al. (2004, 2005) and others. The benefits of using FG-finite element analysis (FEA) over conventional FEA are that: (1) FG does not need a fitted mesh to discretize the analysis domain; (2) the boundary of the design is disassociated from the mesh (Garcia and Steven, 1999); (3) designs using FG-FEA do not contain chequerboard patterns, making the design more reliable for manufacture (Maan et al., 2007); (4) solution time is significantly reduced (Garcia and Steven, 2000).

In FG-FEA, the elements are in a fixed position and have the real design superimposed on them. This means that there are elements that lie inside (I), outside (O) or on the boundary (B) of the design.

The elemental stiffness matrix $\left(\mathbf{K}^{\mathrm{e}}\right)$ is given by Eq. (2):

$$\mathbf{K}^{\mathrm{e}}=\begin{cases}\mathbf{K}_{\mathrm{I}} & \text{if} \quad \xi^{\mathrm{e}}=1\\ \mathbf{K}_{\mathrm{O}}=\mathbf{K}_{\mathrm{I}}\times \mathrm{FG}_{\mathrm{R}} & \text{if} \quad \xi^{\mathrm{e}}=0\\ \mathbf{K}_{\mathrm{B}}=\mathbf{K}_{\mathrm{I}}\xi^{\mathrm{e}}+\mathbf{K}_{\mathrm{O}}\left(1-\xi^{\mathrm{e}}\right) & \text{if} \quad 0<\xi^{\mathrm{e}}<1\end{cases} \tag{2}$$

where ξ^{e} is the design fraction inside the element; $\mathbf{K}_{\mathrm{I}}$, $\mathbf{K}_{\mathrm{O}}$, $\mathbf{K}_{\mathrm{B}}$ are the element stiffness matrices for an element inside, outside, and on the boundary respectively; FG_{R} is the Fixed Grid ratio, usually set in the range of 10^{-6} to 10^{-4}.

The criterion value at the i node $\left(\sigma^{\mathrm{i}}\right)$ is determined by Eq. (3)

$$\sigma^{\mathrm{i}}=\frac{\sum_{\mathrm{e}=1}^{\mathrm{N_e}}\sigma_{\mathrm{e}}^{\mathrm{i}}\times \mathrm{V_e}}{\sum_{\mathrm{e}=1}^{\mathrm{N_e}}\mathrm{V_e}} \tag{3}$$

where σ_e^i is the nodal criterion value at node i for each element (e) surrounding that node, N_e is the number of elements connected to that node, and V_e is the volume for each element that surrounds that node. The nodal value σ_e^i is obtained from the criterion values at each Gauss point extrapolated to the nodes using the shape functions of the element.

The criterion value in each element $\left(\sigma^e\right)$ is calculated using Eq. (4):

$$\sigma^e = \frac{\sum_{i=1}^{N_i} \sigma^i}{N_i} \tag{4}$$

where N_i is the number of nodes of the element.

In the current liteITD program, there are two available finite elements for 2D modelling of solid structures: (1) Quadrilateral; (2) Triangle. The four and three-sided elements include four and three nodes, each with two degrees of freedom (two displacements), and four and one Gauss integration points, respectively. Both elements can be used only as a plane ones (plane stress with thickness).

2.2 Criterion selection

The design criterion (σ) used in liteITD program was the von Mises stress $\left(\sigma_{vM}\right)$, which for a 2D continuum domain is calculated using Eq. (5).

$$\sigma_{vM} = \sqrt{\sigma_x^2 + \sigma_y^2 - \sigma_x \sigma_y + 3\tau_{xy}^2} \tag{5}$$

where σ_x, σ_y and τ_{xy} are the normal and shear stresses, respectively.

2.3 Minimum criterion level extraction

The procedure used to generate the structural boundary depends on the determination of the MCL isoline. To determine the line segments that produce the profile of the boundary, the contouring subroutine called the marching triangles (MT) algorithm (Hinton and Illingworth, 1997) was implemented, since there are no ambiguities and the constructed isolines are smooth.

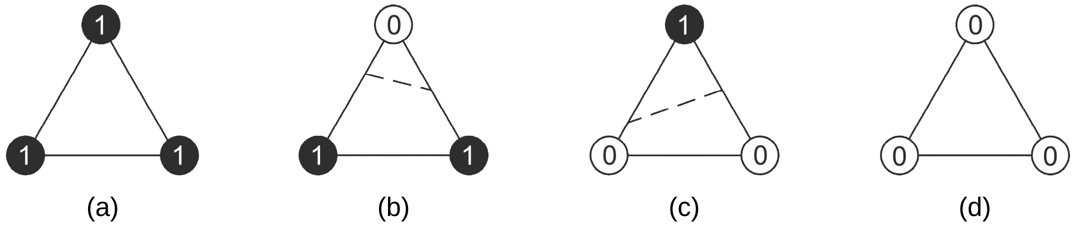

Figure 2. Look-up table for the MT algorithm showing the four different topological states.

The MT algorithm uses a divide-and-conquer approach, treating each finite element independently as a triangular cell. In the case where the element shape is quadrilateral, the element is divided into four triangular cells by introducing a point at the centroid of the element. The criterion value of the central point is obtained by calculating the average values from the four nodes of the original element. The basic assumption of this algorithm is that a contour can only pass through a triangular cell in a limited number of ways. This algorithm requires the value of the MCL together with the value of the criterion at each corner of the cell, and consists of two steps: (1) identify from Figure 2 the topological state of each cell; (2) determine the shape of the contour of the MCL isoline through each cell. The interaction of an isoline through a triangular cell can have a maximum of four different topological states (Figure 2). A value of (1) at a corner means that its criterion value is greater than the MCL, whereas a value of (0) at a corner means that its criterion value is less than the MCL. When the corner in an edge of a cell has different values (0 and 1 or vice versa) it indicates that the MCL isoline intersects that edge, which is the case for topological states shown in Figures 2b and 2c. To find that intersection point, linear interpolation can be used. The shape of the MCL isoline through the cell is then obtained by connecting these intersecting points between the opposite edges as shown in Figures 2b and 2c.

2.4 Structural boundary stabilization

When the MCL is modified, the structural boundary changes and this affects the criterion distribution. Therefore, before the next iteration is started, an iterative process of reanalysis and material redistribution is carried out until the change in the domain volume between successive boundary adjustments is less than a minimum volume change limit (ΔV), given by Eq. (6). Typical values of the minimum volume change limit are around $\Delta V(\%) = 1\%$.

$$\Delta V(\%) = \left(\frac{V_i}{V_{i-1}} - 1 \right) \times 10^2 \quad (4)$$

2.5 Original isolines topology design algorithm

The original ITD algorithm (Victoria et al. 2009) can be explained by the following steps, a schematic representation of which is given in Figure 3.

1. Define the structure: design and non-design domains, material properties, loads and supports.
2. Specify the finite element mesh characteristics.
3. Specify the ITD parameters: design criterion; final design volume; total number of iterations; minimum volume change.
4. Carry out an FG-FEA.
5. Calculate the MCL.
6. Extract the boundary of the structure.
7. If the percentage volume change is greater than the minimum volume change, go to step 8, otherwise go to step 9.
8. Carry out an FG-FEA of the design domain and go to step 6.
9. If the total number of iterations has been reached, go to step 10, otherwise update the design volume, increment the iteration number by 1 and go to step 4.
10. Stop the design process.

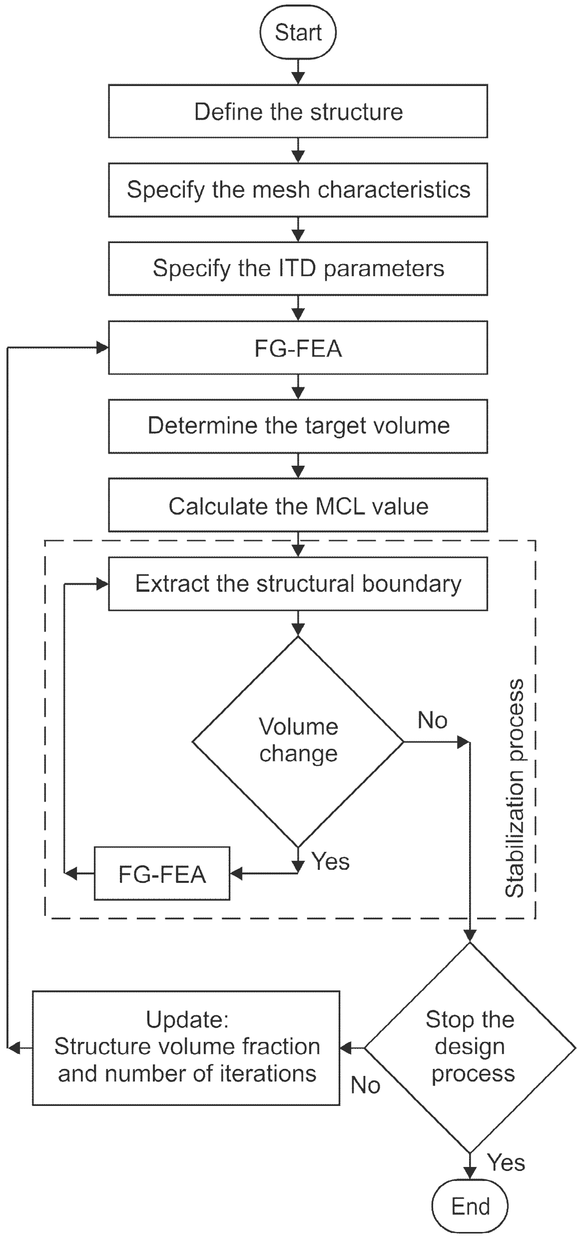

Figure 3. Flowchart showing the steps of the original ITD optimization algorithm.

2.6 Improvements, extension and applications of ITD

Victoria et al. (2009) presented the algorithm for the topological design of two-dimensional structures using isolines called isolines topology design (ITD). The topology and the shape of the design depend on an iterative algorithm, which continually adds and removes material depending on the shape and distribution of the contour isolines of the required structural behaviour. The use of the isolines has two major benefits: (1) although the design criteria can be local (such as stress), by using the MCL isoline to define the shape/topology of the domain, the process works globally; (2) the generated designs have smooth boundaries and need no further interpretation, enhancement or processing.

In this section we introduce a brief review of the ITD method extensions and applications researched in the last years.

Victoria et al. (2011) presented an extension to the original 2D-algorithm for topology design of three-dimensional continuum structures. The novelty of this work was in introducing into ITD the capability of designing 3D structures. The method of determining the isosurfaces was given, together with several examples to show the effectiveness of the algorithm. Victoria et al. (2010) presented an enhancement to the ITD method for topology design under multiple load cases of 2D/3D continuum structures. The results demonstrated how using multiple loading conditions can produce more stable and realistic designs with a little additional complexity. The literature on the topology optimization of truss-like continua has relatively few publications that look at the problem of structures with different material properties behaviour in tension and compression (T&C). The aim of Querin et al. (2010) was to propose an approach suitable for topology optimization of continuum structures with different material properties in tension and compression. Victoria et al. (2011) studied the effect of using different mechanical properties for the steel reinforcement and for the concrete on the emerging topology of strut-and-tie models. The results demonstrated that T&C designs gave the exact location of the reinforcement. However, for the majority of structures, the layout or topology of the steel reinforcement in the T&C designs was pretty different from classic designs. The application of topology optimization to shell structures has been less researched than size and shape optimization. Victoria et al. (2014) introduced into ITD method the capability of designing shell structures to investigate when it is appropriate or not to include the shell membrane or force symmetry in the topology optimization of stiffeners. To obtain the stiffener layout, the shell structure was modelled using overlapping layers of thin-shell FE whose nodes were coupled using multi-point constraints. Querin et al. (2015) presented an enhancement to the ITD algorithm that allows it to produce multi-material designs. The placement of the multiple materials in the design was determined by the distribution of the design criterion (i.e. von Mises stress) following the concept that the material of highest stiffness must support the regions of the design with the highest design criterion.

The current lite version of ITD method includes: (1) single and (2) multiple loading conditions; (3) different material properties/behaviour in tension and compression; (4) multi-material optimization for 2D continuum structures.

3 DESCRIPTION OF THE liteITD SOFTWARE

A user interface (UI) is a graphical display in one or more windows containing controls, called components that enable the user to perform interactive tasks. The user does not have to create a script or type commands at the command line to accomplish the tasks. Unlike coding programs to accomplish tasks, the user does not need to understand the details of how the tasks are performed.

UI components can include menus, toolbars, push buttons, radio buttons, list boxes, and sliders-just to name a few. UIs created using MATLAB tools can also perform any type of computation, read and write data files, communicate with other UIs, and display data as tables or as plots (MathWorks, 2016).

GUIDE is a development environment that provides a set of tools for creating user interfaces (UIs). These tools simplify the process of laying out and programming UIs.

Using the GUIDE layout editor, it is possible to populate a UI by clicking and dragging UI components-such as axes, panels, buttons, text fields, sliders, and so on-into the layout area. The user can also create menus and context menus for the UI. From the layout editor, the user can size the UI, modify component look and feel, align components, set tab order, view a hierarchical list of the component objects, and set UI options.

GUI Options dialog box allows to configure several behaviours that are specific to the GUI that is being created. liteITD GUI was created using the GUIDE tool and it has two main features: MATLAB automatically rescales the components in the GUI in proportion to the new figure window size; MATLAB allows to display only one instance of the GUI at a time.

3.1 System requirements and installation of liteITD

The liteITD program was generated using MATLAB 2014a and it is a 64-bit application for Windows operating system. Although, it has been tested with the newer MATLAB version (MATLAB 2015a).

On the DEyC (Structures and Construction Department) website of the Technical University of Cartagena (UPCT) http://www.upct.es/goe/software/liteITD.php under the directory liteITD, the following three folders may be found:

1. for_redistribution: A folder containing the installer (MyAppInstaller_mcr) to distribute the application.
2. for_testing: A folder containing the raw files (liteITD.exe; mccExcludedFiles.txt; readme.txt) generated by the compiler.
3. for_redistribution_files_only: A folder containing only the files (liteITD.exe; readme.txt) needed to redistribute the application.

Note that liteITD for other operating systems may be issued in the future and the interested users can check the DEyC website for updates.

3.1.1 User with access to MATLAB

liteITD program can be started by double-click on the icon shown in Figure 4. liteITD.exe is the main program of the UI window which allows the user: (1) to build the plane solid model; (2) to create the finite element model; (3) to run the topology optimization.

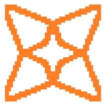

Figure 4. liteITD.exe icon.

3.1.2 User without access to MATLAB

The MATLAB Compiler Runtime (MCR) is a standalone set of shared libraries that enables the execution of MATLAB files on computers without an installed version of MATLAB. Applications that use components built with MATLAB Compiler require access to an appropriate version of the MCR to run. In our case, the installer generated (MyAppInstaller_mcr) by the compiler app include the MCR installer and liteITD application.

To install a MATLAB generated liteITD standalone application, it is necessary to carry out the following steps:

1. Locate the MyAppInstaller_mcr executable in the for_redistribution folder created by the MATLAB Compiler.
2. Double click the liteITD installer (MyAppInstaller_mcr) to advance to liteITD installer (Figure 5).
3. Click Next to advance to the installation options page (Figure 6).
4. Click Next to advance to the required software page (Figure 7). If asked about creating the destination folder, click Yes. If the user already has the correct version of the MCR installed on the system, this page will have a message indicating that the user does not have to install the MCR. If the user receives this message, skip to step 8.
5. Click Next to advance to the license agreement page. If asked about creating the destination folder, click Yes.
6. Read the license agreement and check Yes to accept the license.

7. Click Next to advance to the confirmation page.

8. Click Install. The installer installs the MATLAB generated application.

9. Click Finish.

10. Run your liteITD standalone application.

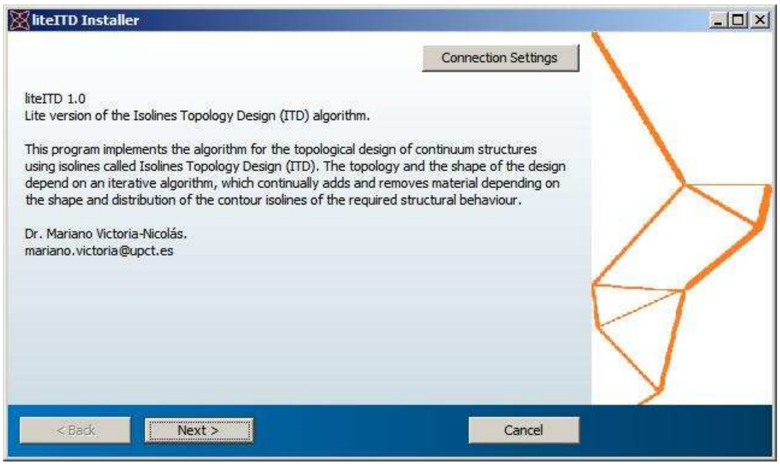

Figure 5. liteITD installation launcher. Application information.

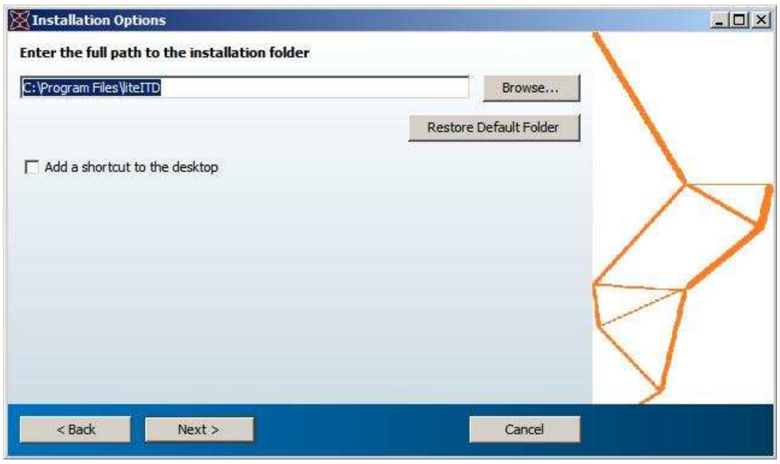

Figure 6. Installation options page.

Figure 7. Required software page.

3.2 Overview of liteITD interface

When liteITD is started, the graphical interface of Figure 8 appears. This is similar to most windows applications as it has most of the usual features. From top to bottom, the interface consists of four sections: (1) Title bar; (2) Menu bar; (3) Button bar; (4) Display area. Users can resize the overall size of the display area. To resize this section, either drag the borders around the window or position the mouse on of the corners of the GUI and drag it diagonally toward the centre of the GUI while holding down the left mouse button. Note that, not all menus and buttons are enabled throughout the work session, it depends on the step in which the user is in the session.

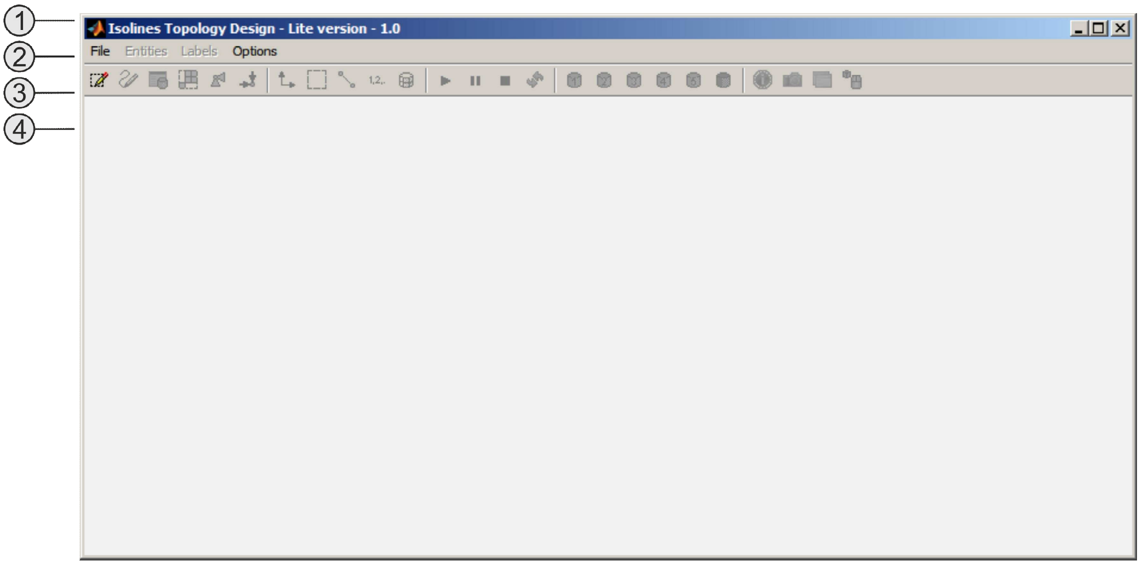

Figure 8. liteITD graphical interface.

3.2.1 Title bar

The Title bar (part 1 in Figure 8) displays the name of the application “Isolines Topology Design – Lite version” together with the version number “1.0”. On the right hand side it also has the standard windows buttons to minimize, maximize, and close the application.

3.2.2 Menu bar

Each menu topic on the Menu bar (part 2 in Figure 8) brings up a pull-down menu of subtopics, which in turn either cascade to a submenu (indicated by a >) or perform an action. The action may do any of the following: (1) immediately execute a command; (2) execute/active a function (indicated by a ✓); (3) bring up a dialog box (indicated by a …). Note that, during a session the user will need to use the menu bar buttons from left-to-right. For this reason, these will switch automatically between enable a disable modes depending on the moment of the work session.

The liteITD workspace consists of all input data in a liteITD session. Remark that, workspace is not maintained across work sessions of liteITD. When the user quits liteITD, the workspace is cleared. However, the user can save all input data to a MAT-file (.mat).

The Menu bar (2) has four menus: Files; Entities; Labels; Options. They are explained in the following sections.

When File is selected, the drop-down list of Figure 9 is displayed. There are six topics in the File menu: Load; Save; Autosave; Change jobname; Change directory; Exit. Each menu topic performs an action.

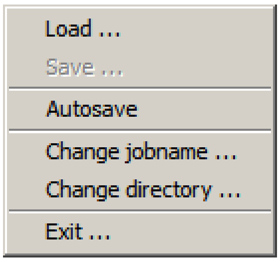

Figure 9. File drop-down menu list.

1. Load: to restore data from the workspace file, select the MAT-file in the current folder browser, left-click, and then select open. The user can load or resume MAT-files in any moment of the work session. A resume operation replaces the data currently in memory with the data in the named workspace file.
2. Save: to save all input data in a file called jobname.mat. It is a good practice to save the job at different times throughout the building of the model to backup the work in case of a system failure or other unforeseen problems.
3. Autosave: if autosave feature is on (indicated by a ✓). Automatically, liteITD saves in each step, the workspace in a file called "jobname_asv-step.mat". Where "step" is a label that helps to understand in which point of the work session was saved the workspace. There are six step labels: wb (workbench); geo (geometry); mat (material properties); mesh (mesh of finite elements); sup (support conditions); for (forces).

The "jobname" is a name that identifies the liteITD job. When user defines a jobname, this becomes the first part of the name of all files the session creates. By using a jobname for each work session, user insures no files are overwritten. When user does not specify a jobname, all files receive the name file.

4. Change jobname: allows the user to specify the jobname. Type the new jobname in the available field of the dialog box (Figure 10).

Figure 10. Dialog box used by liteITD to change jobname.

"Working directory" is a specific folder for liteITD to store all of the files created during a work session.

5. Change directory: allows the user to set the folder (by file selection dialog box) in which the liteITD run will be executed. The initial name of this folder defaults to outITD, and it can be found by default in the folder where the liteITD engine is.
6. Exit, will exit the program (Figure 11).

Figure 11. Dialog box for exiting the liteITD.

When “Entities” is selected, the drop-down list of Figure 12 is displayed. There are seven topics in the Entities menu: Point (points); Line (lines), Area (areas); Node (nodes of the finite elements); Element (finite elements); Support (support conditions); Force (forces).

Figure 12. Dialog box for selecting entities to draw and for labels to show.

When an entity is enabled, it is selectable in the normal manner. When it is disabled (there are no entities of that type), it will be greyed out. Clicking on the enabled items (checked items have a tick mark against them), user can select those entities of the model that user wants to draw.

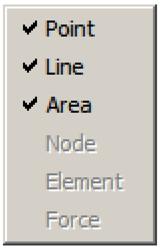

Figure 13. Dialog box for labels to show.

When “Labels” is selected, the drop-down list of Figure 13 is displayed. There are six topics in the Labels menu: Point (point numbers, at the right of items); Line (line numbers, at the middle of items); Area (area numbers, at the centre of items); Node (node number, at the right of items); Element (finite element numbers, at the centre of items); Force (force values written at the right of arrows).

Clicking on the enabled items, user can control those entity labels that user wants to plot.

When “Options” is selected, the drop-down list of Figure 14 is displayed. There are four topics in the Options menu: Stiffness matrix; Designs; Results per; Optimization diary.

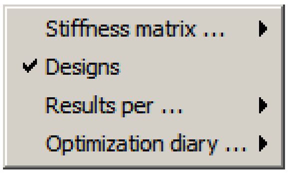

Figure 14. Options drop-down menu list.

When “Stiffness matrix” is selected, the submenu of Figure 15 is displayed. There are two options: (1) Dense (full matrix); (2) Sparse (sparse matrix).

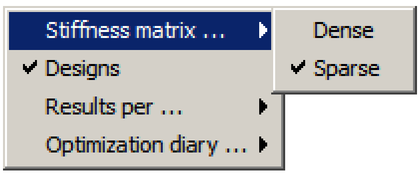

Figure 15. Stiffness matrix submenu.

It is not uncommon to have matrices with a large number of zero-valued elements and, because the MATLAB software store zeros in the same way it stores any other numeric value, these elements can use memory space unnecessarily and can sometimes require extra computing time. Sparse matrices provide a way to store data that with a large percentage of zero elements more efficiently. While full matrices internally store every element in memory regardless of value, sparse matrices can significantly reduce the amount of memory required for data storage. For this reason, it is the checked option by default (Figure 15).

When “Designs” is checked (option by default, Figure 15), the resulting designs shown in display area during the optimization are saved to a graphic file (bmp file format) called “jobname_it-%it_sIt-%sIt.bmp”, where “jobname” is the jobname, “%it” is the iteration number and “%sIt” is the stabilization number (or subiteration).

“Result per” allows the user to fix which is the frequency to save (in a graphic file) and display (on the screen) the resulting designs. When Result per is selected, the submenu of Figure 16 is displayed. There are two frequency options: (1) Iteration; (2) Subiteration (option by default).

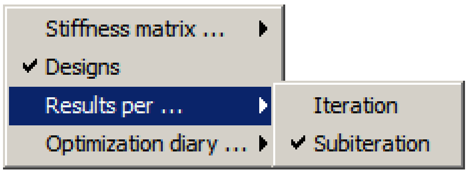

Figure 16. Result per submenu.

“Optimization diary” allows the user to export the results of the optimization to file named jobname_optEvol. When Optimization diary is selected, the submenu of Figure 17 is displayed. There are two file format options: (1) *.csv (comma separated values text files); (2) *.txt (delimited text files, in which the TAB character typically separates each field of text).

Note that, if the optimization result file is a csv file (option by default), Microsoft Excel (or similar programs) automatically opens the file and displays the data in a new workbook. On the contrary, if the file is a text file (.txt), Excel starts the Import Text Wizard.

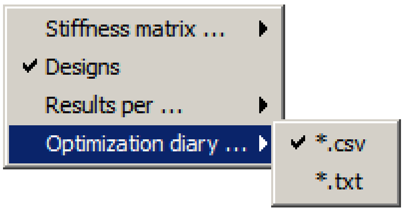

Figure 17. Optimization diary submenu.

After the optimization, the optimization result file can be found in the working directory. Part of a result file in txt file format is depicted in Figure 18.

```
It sIt nAnl timeOpt frVol maxDX maxDY minVM meanVM maxVM
1 0 1 5.243293e-01 0.9817 -5.913135e-07 -1.930125e-06 1.145839e+03 4.074569e+04 1.404917e+05
1 1 2 1.319950e+00 0.9802 -5.852926e-07 -1.736785e-06 8.987488e+03 4.238846e+04 1.404917e+05
2 0 3 1.937809e+00 0.9618 5.851441e-07 -1.736824e-06 8.913492e+03 4.238598e+04 1.404917e+05
2 1 4 2.435620e+00 0.9575 5.811636e-07 -1.743623e-06 8.980456e+03 4.264713e+04 1.404917e+05
3 0 5 2.971723e+00 0.9383 -5.747266e-07 -1.744528e-06 1.394023e+04 4.324693e+04 1.404917e+05
3 1 6 3.601063e+00 0.9308 4.546590e-07 -1.740759e-06 1.197101e+04 4.366425e+04 1.405925e+05
4 0 7 4.111499e+00 0.9087 4.437910e-07 -1.612558e-06 1.649981e+04 4.424509e+04 1.406430e+05
4 1 8 4.714457e+00 0.8900 4.305871e-07 -1.619818e-06 1.606383e+04 4.479970e+04 1.418084e+05
4 2 9 5.238792e+00 0.8742 4.602203e-22 -1.489171e-06 1.716913e+04 4.569078e+04 1.437429e+05
4 3 10 5.873483e+00 0.8588 -3.229694e-07 -1.499056e-06 1.829609e+04 4.649300e+04 1.455201e+05
```

Figure 18. Part of a result optimization file.

Here the results of the first four iterations are presented. The first and second column of data identifies the iteration (It) and subiteration (sIt) numbers, respectively. The third column displays the total number of finite element analysis (nAnl). The fourth column shows the total time spent by the optimization. The fifth column indicates the design volume fraction. The last five columns present the values of maximum displacement in X- and Y-directions (maxDX, maxDY); minimum, mean, maximum von Mises stresses (minVM, meanVM, maxVM) for those nodes which lie inside or on the boundary of the resulting designs.

In case of multiple loading conditions or different material properties in tension and compression the last five column are not written in the result file.

3.2.3 Toolbar area

The toolbar area (part 3 in Figure 8) contains five sections: main toolbar; view toolbar; run toolbar; material toolbar; miscellaneous buttons. When the user moves the mouse pointer onto any button in the Toolbar area, a tooltip string will pop up to give an explanation of the button functions. This section will explain what each button does.

Main toolbar

The main toolbar is depicted in Figure 19 and contains six buttons as explained below.

Figure 19. The main toolbar.

This is the “Define workbench dimensions” button . By pressing this, it opens the dialog box of Figure 20, and allows the user to specify the width and height of the workbench. The liteITD program does not assume a system of units for your work session. The user can use any system of units (units must be consistent for all input data).

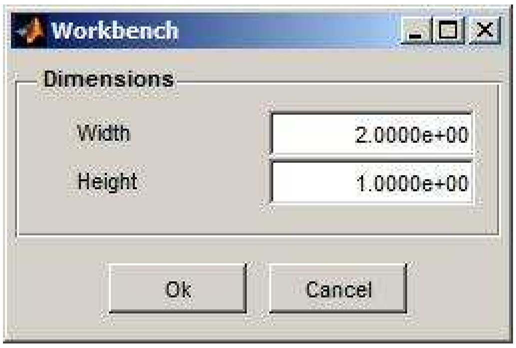

Figure 20. Dialog box for defining the workbench dimensions.

After the workbench dimensions are defined, the user can draw the geometry of the model.

This is the “Draw geometry” button . By pressing this, it opens the “Geometry” dialog box shown in Figure 21, and allows the user to draw the geometric model of the structure to optimize.

To obtain an analysis model is necessary to generate a finite element model (nodes and finite elements). There are mainly two strategies to create the analysis model: (1) direct generation; (2) solid modelling. With direct generation, the user manually defines the location of each node and the connectivity of each element. With solid modelling, one describes the geometric model, then instructs the program to automatically mesh the geometry with nodes and elements.

liteITD makes use of solid modelling strategy to relieve user of the time-consuming task of building a complex analysis model by direct generation. Take into account any plane geometric model is defined in terms of points, lines, and areas. Points are the vertices, lines are the edges, and areas are the faces. There is a hierarchy in these entities: areas, the highest-order entities, are bounded by lines, which in turn are bounded by points.

“Pick mode” allows the user to add, select, and unselect an entity or location by clicking on it in the display area. To enable or disable pick mode the user must click the check box called “By mouse click on screen” shown in Figure 21. A summary of the mouse-button assignments used during a picking operation is given below:

1. Left button adds the entity or location closest to the cursor of the mouse.
2. Right button selects the entity or location closest to click location.
3. Double-click any mouse button unselects the picked entities.

liteITD allows the user to create (“Add” button), remove (“Delete” and “Clear” buttons), and select (list boxes) points, lines, and areas. To manage these entities the user must use the radio buttons “Point”, “Line”, and “Area”.

1. Point (by coordinates): (1) by typing the coordinates in the edit boxes (X- and Y-coordinate, Figure 21) and then pushing “Add” button; (2) by picking in the display area (point coordinates in real time are shown in the edit boxes) with the left button of the mouse (picking mode must be enabled, Figure 21). The user can use the point list (Figure 21) for selecting a point: (1) clicking in the point list with left button

of the mouse; (2) picking on the screen with the right button of the mouse. When a point is selected, this is remarked in the point list and on the screen (using a square marker of blue colour). “Delete” button removes the selected point from the list, and “Clear” button removes all points. In case of remove a point, bounded lines and areas are also removed. When any entity is deleted a reordering procedure is made to avoid numeration jumps.

2. Line (from two points): (1) by typing the “Start” and “End” points in the edit boxes (Figure 21) and then pushing “Add” button; (2) by picking in the display area with the left button of the mouse. The user can use the line list (Figure 21) for selecting a line: (1) clicking in the line list with left button of the mouse; (2) picking on the screen with the right button of the mouse (at the middle of line). When a line is selected, this is remarked in the line list and also on the screen (using a ticker blue line). “Delete” button removes the selected line from the list, and “Clear” button removes all lines. If a line is removed, bounded areas are also removed.
3. Area (by lines): (1) by typing “Line/s” in the edit box (Figure 21) and then pushing “Add” button; (2) by picking in the display area with the left button of the mouse (at the centre of area) and then pushing “Add” button. The user can use the area list (Figure 21) for selecting an area: (1) clicking in the area list with left button of the mouse; (2) picking on the screen with the right button of the mouse. When an area is selected, this is remarked in the area list and also on the screen (using a blue background). “Delete” button removes the selected area from the list, and “Clear” button removes all areas. Note that, (1) lines contained in areas do not need to be specified in consecutive order (although, in the area list they are sorted in counter clockwise). (2) When number of lines to input is high, MATLAB provides a special shortcut notation for this circumstance using the colon operator. This operator specifies a whole series of values by specifying the first value (first) in the series, the stepping increment (incr), and the last value (last) in the series. For example,

the expression 1:2:8 (first:incr:last) is a shortcut for a 1×4 row vector containing the values 1,3,5, and 7.

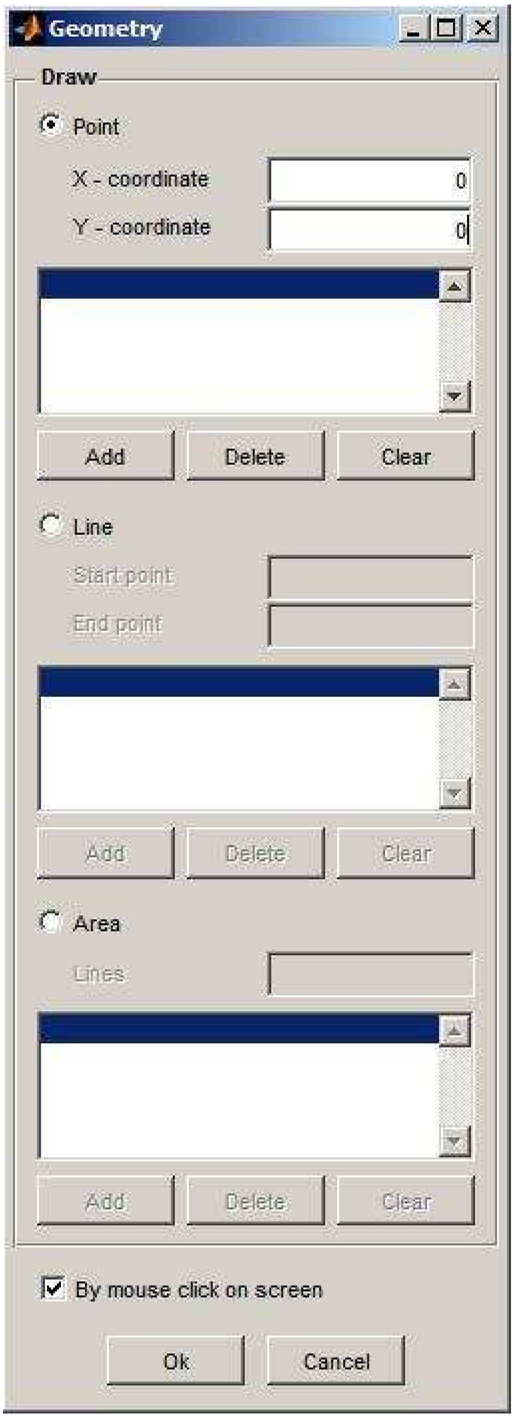

Figure 21. Dialog box for drawing the geometric model.

After drawing the geometric model, the user can specify the mechanical properties of the materials.

The "Define material properties" button enables the user to specify the mechanical properties of the materials for the analysis model. By clicking on this button, a dialog box shown in Figure 22 will appear where the user can input Young's modulus and Poisson's ratio for each material. The liteITD program allows to define up to five material types, by clicking the pop-up menu labelled "Number of materials". Each set of material properties has a material reference number ("No.") and can be differentiated by colours (cyan, black, magenta, yellow, and green are available). Take into consideration: (1) the unit system must be consistent; (2) Poisson's ratios must be in the range of 0 to 0.5; (3) Young's moduli must be positive.

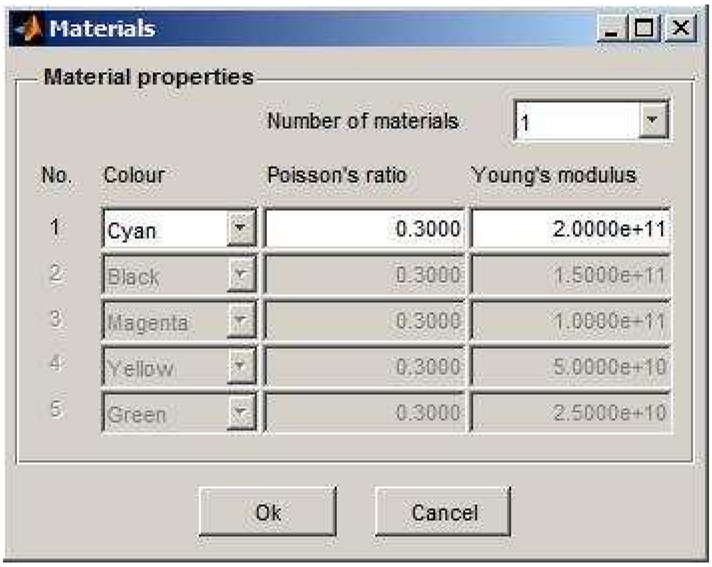

Figure 22. Dialog box for specifying the material properties.

After the material properties are defined, the geometric model is ready to be meshed in the following step.

By clicking on this button in the main toolbar, the "mesh" dialogue shown in Figure 23 will appear. The dialog box is divided in five sections: (1) Fixed grid; (2) Shape; (3) Type; (4) Size; (5) Attributes.

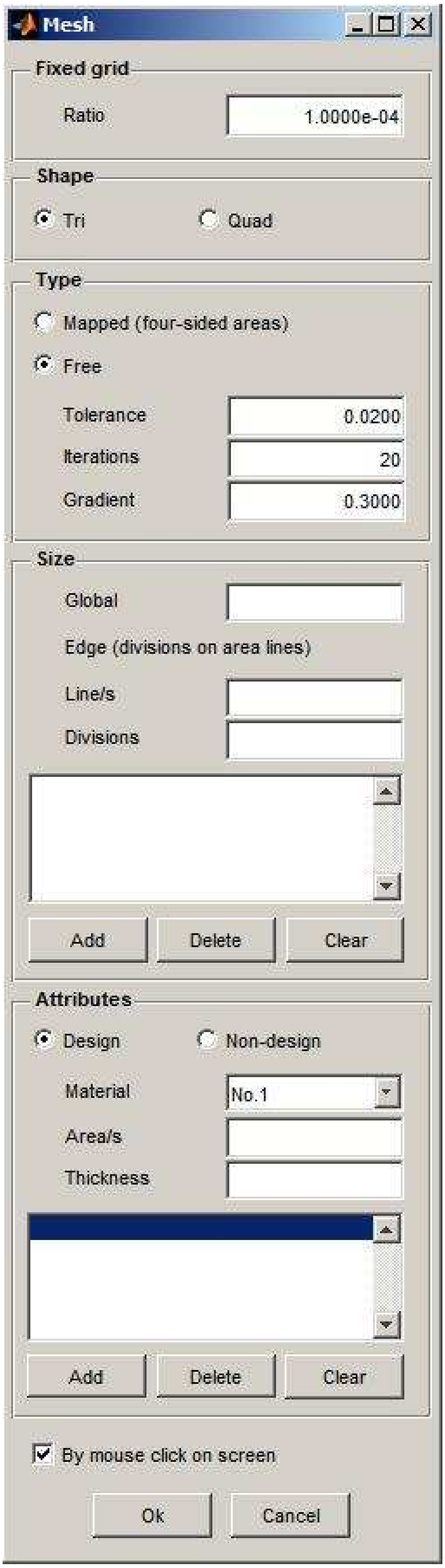
Mesh
Fixed grid
Ratio
1.0000e-04
Shape
Tri
Quad
Type
Mapped (four-sided areas)
Free
Tolerance
0.0200
Iterations
20
Gradient
0.3000
Size
Global
Edge (divisions on area lines)
Line/s
Divisions
Add
Delete
Clear
Attributes
Design
Non-design
Material
No.1
Area/s
Thickness
Add
Delete
Clear
By mouse click on screen
Ok
Cancel

Figure 23. Dialog box where the features of the fixed grid of finite elements are specified.

1. Fixed grid: in this section, the user can type in the edit box the FG_R value (Eq. 2), which defaults to $1{\times}10^{-4}$.
2. Shape: in this section the finite element shape is set. To specify the desired shape, one can use the labelled radio buttons "Tri" and "Qua" for triangular- and quadrilateral-shaped elements, respectively. Note that, the mixture of the two shapes in the same model is not allowed.
3. Type: in addition to specifying element shape, the user can also set the type of meshing (free or mapped) that must be used to mesh the geometric model. To specify the mesh type, the user can use the radio buttons called "Mapped" and "Free". Before meshing the model, and even before building the model, it is crucial to think about whether a free or mapped mesh is appropriate for the FEA and for the optimization.

   When mapped option is selected, the user must build the geometry as a series of fairly regular areas that can accept a mapped mesh. For an area to accept a mapped mesh, two conditions must be satisfied: (1) the area must be bounded by four lines (four-sided areas); (2) the area must have equal number of element divisions specified on opposite sides. Note that, (1) liteITD can only create a mapped mesh within convex 4-sided regions; (2) liteITD generates mapped triangle mesh starting from mapped quadrangle mesh, dividing each quadrangle into two triangles.

   In free meshing operations, no special requirements restrict the solid model. Any geometry, even if this is irregular, can be meshed. When free option is selected, the edit boxes called "Tolerance", "Iterations", and "Gradient" are active. These options allows the user to modify the default behaviour of the free mesh generator, where "Tolerance" is the converge tolerance (the maximum relative change in edge length per iteration must be less than this value, which defaults to 0.02), “Iterations” is the maximum allowable number of iterations, which defaults to 20, and “Gradient” is the maximum allowable (relative) gradient in the size function, which defaults to 0.3. Note that, liteITD generates free quadrangle mesh starting from free triangle mesh by quadrangulation (by joining the triangle centroids to the midpoints of its sides, three quadrilaterals are obtained for each of these triangles).

4. Size: controls the element size used to create the mesh. The user can use either “Global” option to specify the maximum allowable global element size (maximum edge length), or “Edge” option to specify the number of divisions on lines (shortcut notation is allowed), or both options together. In that case, liteITD will use the option that provides the smallest element size. The list box of this panel group allows the user to select lines or manage (add, delete, or clear) the size specifications.
5. Attributes: before generating a mesh, one must first define the appropriate area attributes. That is, it must be specified the following: (1) “Design” or “non-design” domain (by radio buttons). Remark that, a non-design domain is a region that cannot be optimized; (2) “Material” properties set (by pop-up menu); (3) “Area/s” (by edit box, shortcut notation is allowed); (4) “Thickness” of the “Area/s” (by pop-up menu). The list box of this panel group allows the user to select areas or manage (add, delete, or clear) the area attributes.

A summary of the mouse-button assignments when pick mode is active is given below:

1. Left button adds to “Line/s” edit box line closest to the mouse cursor.
2. Right button adds to “Area/s” edit box area closest to the mouse cursor.
3. Double-click any mouse button unselects any picked entity.

Clicking on the “Ok” button will dismiss the dialog box and the liteITD will start to generate a mesh. In case of free meshing operations a wait bar dialog box of Figure 24 appears. This bar displays what percentage of a meshing is complete by progressively filling a bar with red from left to right.

Figure 24. Wait bar that displays the progress of free meshing operations. The message text shows the meshing area number and the number of completed iterations.

After the geometric model is meshed, the user can apply the support conditions.

This is the "Input support conditions" button . By pressing this, the "Displacements on nodes" dialog box of Figure 25 appears, and allows the user to specify constrained displacements on nodes. The dialog box is divided in two sections: (1) Constrained freedom; (2) Selection.

1. Constrained freedom: liteITD allows the user applies only DOF (Degree Of Freedom) constraints on nodes. The user can choose from three DOF constraint conditions: (1) Displacement in X-direction is fixed (X-displacement box is checked); (2) Displacement in Y-direction is fixed (Y-displacement box is checked); (3) All displacements are fixed (X- and Y- displacement boxes are checked, Figure 25).
2. Selection: the user can select a node by keyboard entry (by typing the node number in the "Node/s" edit box, shortcut notation is allowed) or by graphical picking (pick mode must be active). For retrieval picking (identify existing identities), the user can choose among "Single", "Box", and "Polygon" mode. In single pick mode, each click on the left mouse button picks one node. With box and polygon modes, the user can press and drag the mouse to enclose a set of nodes in a box or polygon. Notice that, when a node is selected, this is remarked using a square marker of blue colour.

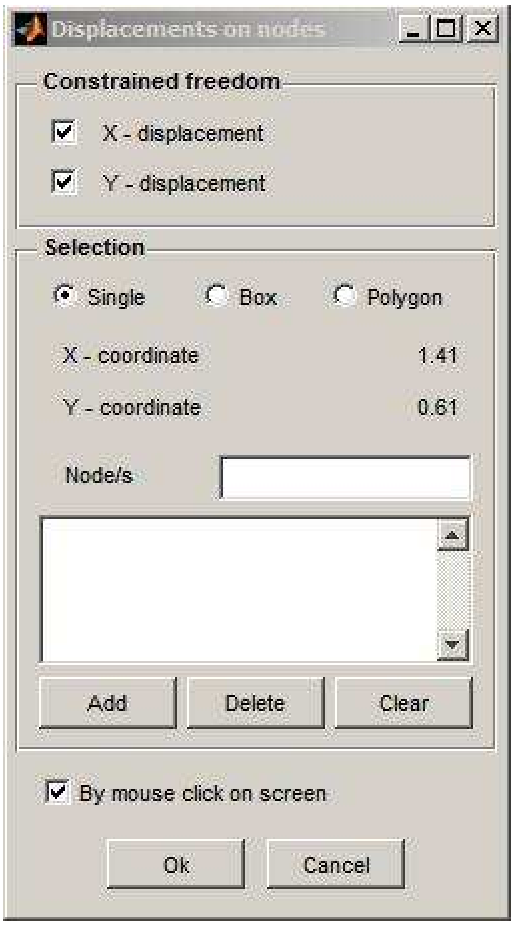

Figure 25. Dialog box for specifying constrained displacements on nodes.

When the “Add” button is pressed the DOF constraints specified above are applied to selected nodes and shown in the list box. “Delete” button removes the constrained node from the list, and “Clear” button remove all constrained nodes.

A summary of the mouse-button assignments when pick mode is active is given below:

1. Left button adds to “Node/s” edit box the node closest to the mouse cursor.
2. Right button selects (from nodes shown in the list box) node closest to the mouse cursor.
3. Double-click any mouse button unselects all picked nodes.

Notice that, when pick mode is active the mouse cursor coordinates are shown, in real time, in “X-coordinate” and “Y-coordinate” edit boxes.

After the support conditions are specified, the user can apply the loading conditions.

This is done by means of the “Input nodal forces” button . By pressing this, the “Forces on nodes” dialog box of Figure 26 appears, and allows the user to apply forces (concentrated loads) on nodes. The dialog box is divided in three sections: (1) Load case number; (2) Nodal force; (3) Selection.

1. Load case number: the user must specify the number of load case in this edit box. When the button “Apply” is pressed, “Nodal force” and “Selection” panels are activated.
2. Nodal force: the user can choose which load case is current by left-clicking on the “Load case” pop-up menu. In “X-direction” and “Y-direction” edit boxes, the user can specify the force values that will be applied to selected nodes in the model. Be aware that the unit system be consistent.
3. Selection: This panel works equal to “Selection” section from “Displacements on nodes” dialog box.

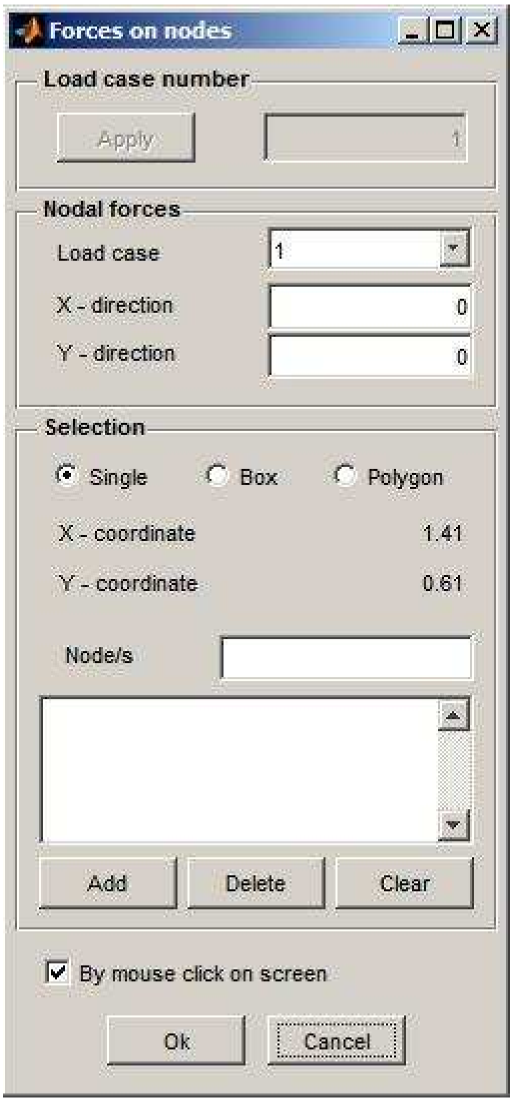

Figure 26. Dialog box for specifying loading conditions.

View toolbar

The view toolbar is depicted in Figure 27 and contains five buttons as explained below.

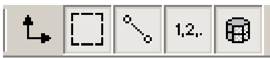

Figure 27. The view toolbar.

This is the “Show triad” button . When this button is in pressed down position, it shows the global XY coordinate triad on display (in lower left screen corner).

This is the “Show workbench” button . When this button is in pressed down position, it shows the workbench dimensions through a dashed black rectangle.

This is the “Show entities” button . When this button is in pressed down position, it shows the entities marked with tick symbol in the “Entities” menu.

This is the “Show labels” button . When this button is in pressed down position, it shows the labels marked with tick symbol in the “Labels” menu.

This is the “Display edges of finite elements” button . When this button is in pressed down position, it displays the boundaries (edges) of finite elements with a solid black line.

Run toolbar

The run toolbar is shown in Figure 28 and contains four buttons as explained below.


Figure 28. The run toolbar.

This is the “Optimization launcher” button . By clicking on this button, the “Optimization” dialog box of Figure 29 appears, and allows the user to specify the liteITD parameters. The values shown in this figure are default settings specified by liteITD for all designs. The dialog box is divided in four sections: (1) Material; (2) Load case; (3) T-ension/C-ompression, (4) Evolution.

1. Material: the user can choose among single or multi-material design. When “Single” option is selected the user must specify a material from pop-up menu. On the other hand, if multiple material option is selected the user must click which will be the materials and the volume control weighting factors used to optimize the structure (Querin et al. 2015). Notice that, when there is an only material, the “Multiple materials (Fractions)” option will be disabled.
2. Load case: the user can choose among single or multiple loading conditions. When “Single” option is selected the user must specify a load case from pop-up

menu. Otherwise, the optimization is carried out for multiple load cases (Victoria et al. 2010). Notice that, when there is only a single load defined, the "Multiple" option will be disabled.

3. T-ension/C-ompression: if the user desires a topology optimization with different material properties in tension and compression "T/C-material properties" option should be selected (Querin et al. 2010). In that case, density of tension and compression members (T-density, C-density) and Young's moduli of tensile and compressive members (T-Young's mod., C-Young's mod.) must be specified. The "T/C colours" option allows the user to display the regions in tension and compression in red and blue colours, respectively.
4. Evolution: the evolutionary parameters include number of iterations ("Iterations"); minimum volume change, in percentage ("Min. change (%)"); Objective volume fraction, in percentage ("Obj. fraction (%)").

Notice that, the topology optimization is allowed only in the following cases: (1) single material, single load case, no differentiation T/C; (2) single material, single load case, different material properties in tension and compression; (3) single material, multiple loading conditions, no differentiation T/C; (4) multi-materials, single load case, no differentiation T/C.

Clicking on the "Ok" button in the dialog box will start the liteITD optimization process. The computational cost of the optimization will depend mainly on size of finite element model and number of load cases.

This is the "Optimization pause" button ⏸. Pressing this button, the optimization process is stopped until the user presses the button again.

This is the "Optimization stop" button ■. Pressing this button, the optimization process is stopped.

This is the "Restart liteITD program" button . By clicking on this button, all input data are deleted and liteITD program is restarted.

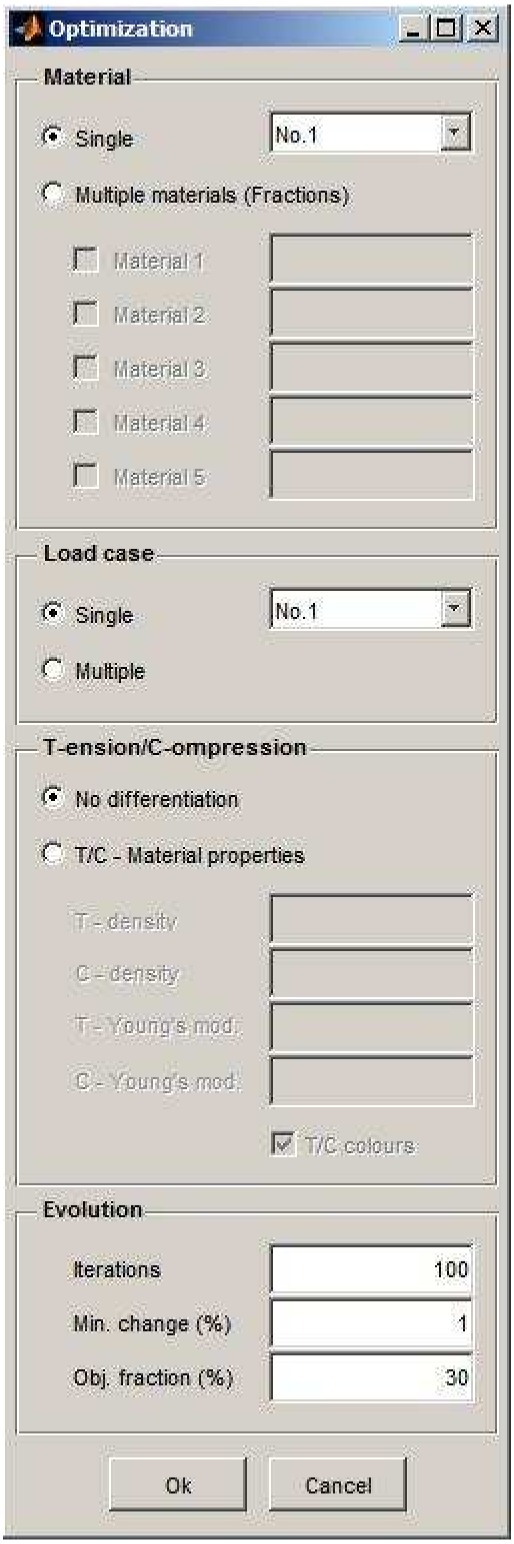

Figure 29. Dialog box for defining liteITD parameters.

Material toolbar

The material toolbar is shown in Figure 30 and contains six buttons. The first five button (from left to right) allows the user to filter by material in design domain. The last button allows to show/hide the regions treated as non-design domain in the resulting designs. Notice that, (1) the number of enabled buttons to filter by material is consistent with the material number used in the optimization; (2) the "Display non-design regions" is enabled whether there is any region of this type in the model.

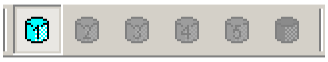

Figure 30. The material toolbar.

Miscellaneous buttons

The miscellaneous button toolbar is shown in Figure 31 and contains four buttons as explained below.

Figure 31. The miscellaneous button toolbar.

This is the "Show information" button . When this button is in pressed down position, the liteITD shows on the display area (Figure 32) relevant information for the user:

On the left side of the screen (from top to bottom, Figure 32a): working directory (Working directory); jobname (Jobname) ; time; date; number of points (Points), lines (Lines), areas (Areas), and materials (Materials); fixed grid ratio (FG ratio); number of nodes (Nodes), elements (Elements), degree of freedoms (DoFs), constrained degree of freedoms (Const. DoFs); number of load cases (Load cases); number of materials (Opt. materials) and load cases (Opt. load cases) that take part in the optimization; tension and compression factor (T/C factor); number of iterations (Iterations); minimum volume change (Min. change); objective volume fraction (Obj. Fraction).

On the right side of the screen (from top to bottom, Figure 32b): current iteration (Iter.) and subiteration (SubIter.); number of finite element analysis (FE analysis); elapsed time (Time); current volume fraction (Vol. frac.); number of non-design elements (ND elem.), inside (I elem.), (O elem.) outside, and boundary (B elem.) elements; maximum displacement in X- (Max. dx) and Y- (Max. dy) directions; minimum (Min. vM), mean (Mean vM), and maximum (Max. vM) von Mises stress in nodes which lie inside or on the boundary of the design.

Working directory: C:\liteITD\outITD
Jobname: file
12:31:01
14-Jul-2015
Points:4
Lines:4
Areas:1
Materials:1
FG ratio:1.0000e-04
Nodes:280
Elements:243
DoFs:560
Const. DoFs:20
Load cases:1
Opt. materials:1
Opt. load cases:1
T/C factor:1.0000
Iterations:100
Min. change:0.0100
Obj. fraction:0.3000

(a)

Iter.:3
SubIter.:1
FE analysis:4
Time:1.57e+01
Vol. frac.:0.9684
ND elem.:0
I elem.:227
O elem.:4
B elem.:12
Max. dx:1.1434e-08
Max. dy:-6.4768e-08
Min. vM:2.7096e+02
Mean vM:1.5349e+03
Max. vM:5.6677e+03

(b)

Figure 32. Information shown on the display area during the optimization.

Note that, depending on the moment of the work session and the optimization options, part of this information can be hidden.

This is the "Screenshot" button. This button allows the user to save the display area to a ".bmp" file called "jobname_screenshot_%iShot", where "jobname" is the jobname and "%iShot" is the capture number.

This is the “Raise hidden menu” button . By clicking on this button the user can raise hidden windows to the top of liteITD graphical interface.

This is the “Active dynamic model manipulation” button . Pressing this button activates dynamic model manipulation mode. The user can use the mouse pointer to pan or zoom the model. Press and hold down the left mouse button to pan the model. Mouse scroll wheel lets user make zoom in/out. Double-click with any mouse button resets the view to initial view.

Notice that, when the “By mouse click on screen” check box (see Geometry, mesh, displacements and forces on nodes pop-up menus) is checked the dynamic model manipulation model is disabled.

4 RUNNING liteITD FROM GUI

This section has been written to guide, step-by-step, how to use the liteITD GUI window to construct a model and then optimize it. The procedure includes the following steps:

1. Defining the design workbench dimensions.
2. Drawing the geometric model.
3. Specifying the material properties.
4. Generating a finite element mesh.
5. Applying DOF constraints.
6. Applying loading conditions.
7. Specifying liteITD parameters.
8. Viewing the resulting optimal design.

To show how liteITD works, we will optimize a typical Michell type structure (Michell, 1904). This example is the short cantilever shown in Figure 33. The design domain was a rectangular area of 1.5×1 m and the thickness was 0.01 m. The mesh used for the discretization of the design domain had 150×100 elements. A vertical force of 1000 N was applied at the centre of the free end in the downward direction and the cantilever was fully clamped along the left edge. The Young's modulus of all elements was 2×10^{11} Pa, Poisson's ratio was 0.3, and fixed grid ratio was 1×10^{-4}. The ITD parameters were: total number of iterations 100; final volume fraction 10%; minimum volume change 1%.

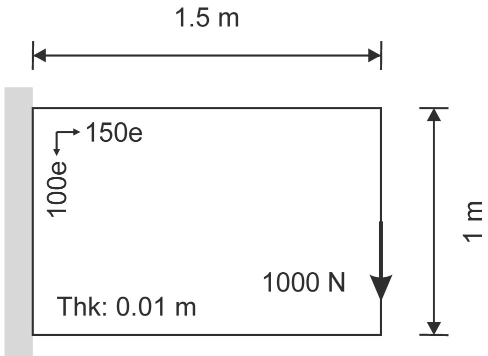

Figure 33. Design domain for the short cantilever loaded at the centre of the free end.

1. Start liteITD.exe.
2. Click on "Autosave" subtopic (GUI: Menu bar>File>Autosave).
3. Open "Change jobname" subtopic (GUI: Menu bar>File>Change jobname…). When the dialog box of Figure 10 appears, type "Short_Cantilever_SLC" and press "OK" (Figure 34).

Remark that, these two last actions are not strictly necessary but are strongly recommended.

4.1 Defining the design workbench dimensions

This step involves the next actions:

1. Press the "Define workbench dimensions" button from the main toolbar.
2. When the "Workbench" dialog box of Figure 20 appears, enter "1.5" in the "Width" edit box, then press "Ok" (Figure 35).

3. Press the “Show information” button from the main toolbar. The resulting workbench is drawn in the display area how is shown in Figure 36.

Figure 34. Change jobname dialog box showing the new jobname.

Figure 35. Workbench dialog box showing the workbench dimensions.

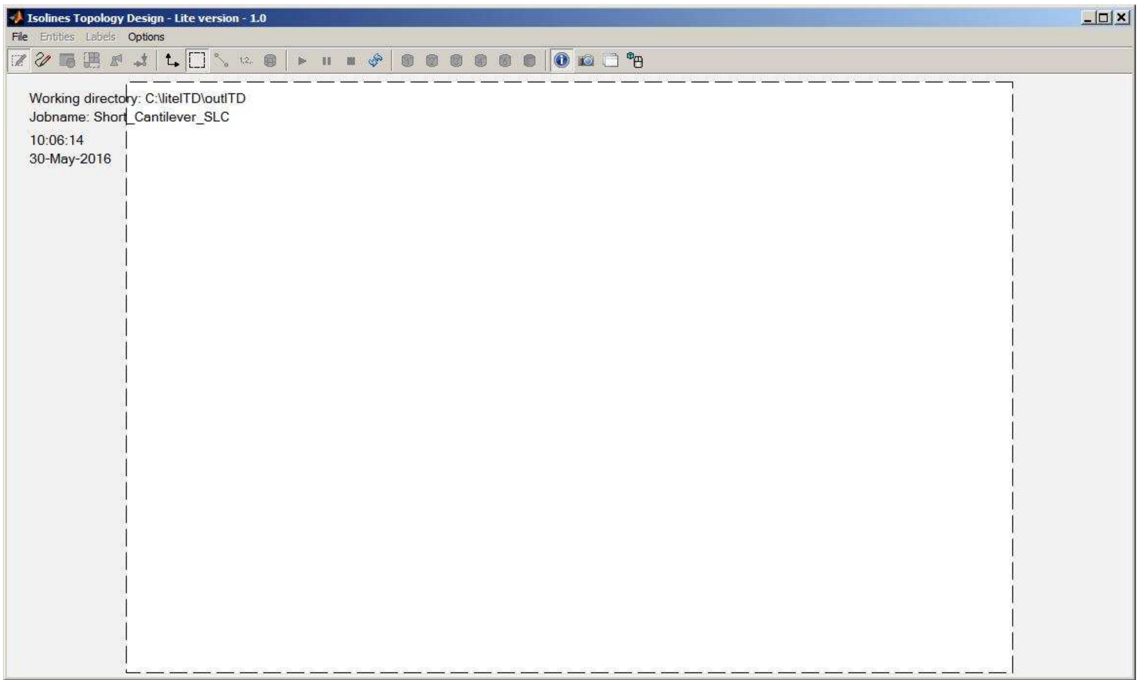

Figure 36. liteITD graphical interface showing drawn workbench.

4.2 Drawing the geometric model

After having defined a workbench, follow the next actions to draw the geometric model. In this case, a rectangular-shaped area:

1. Press the “Draw geometry” button in the main toolbar. The “Geometry” dialog box of Figure 21 will appear. Note that, when the user creates/deletes entities such as points, lines, or areas, these are automatically numbered.
2. Pick the location of the corners of workbench or type the point coordinates as follows: enter “0” in the “X-coordinate” and “0” in the “Y-coordinate” edit boxes, then press “Add”. Repeat the procedure with the remaining points of coordinates (x, y): (1.5, 0); (1.5, 1); (0, 1).
3. Select the “Line” radio button. Pick the points “1” and “2” or type 1 in the “Start point” and 2 in the “End point” edit boxes, then press the “Add” button. Repeat the procedure with the point couples: “2” and “3”; “3” and “4”; “4” and “1”.
4. Select the “Area” radio button. Pick the lines “L1”, “L2”, “L3”, and “L4” or type e.g. “1:4” in the “Line/s” edit box, then press the “Add” button (Figure 37). The grey area is numbered as “A1” (Figure 38).
5. Press “Ok”. The expected geometry is shown in Figure 38. The “dynamic model manipulation” mode can be activated to fit the view.

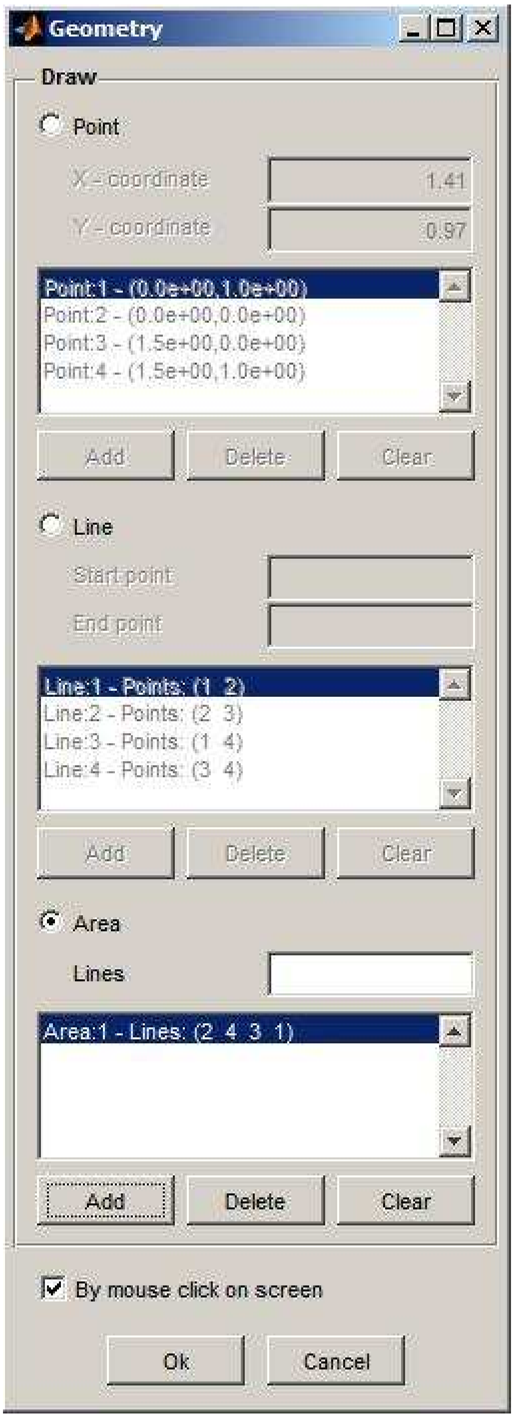

Figure 37. Geometry dialog box showing the geometry data.

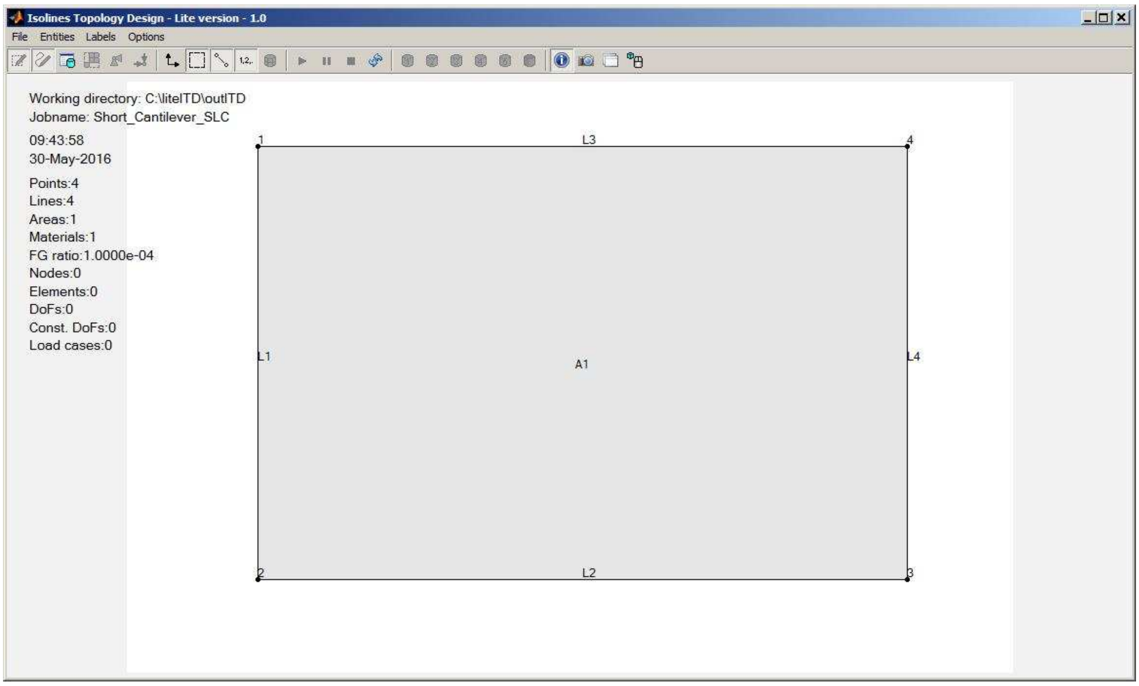

Figure 38. liteITD graphical interface showing drawn geometric model.

4.3 Specifying the material properties

Press the “Define material properties” button located in the main toolbar and the “Materials” dialog box of Figure 22 will appear with those same default values. For this example, the default values need not be changed. Just press the “Ok” button.

4.4 Generating a finite element mesh

After the material properties are defined, the geometric model is ready to be meshed in this step. From within the main toolbar, press the “Mesh generator” button. The “Mesh” dialog box of Figure 23 will appear.

1. From the “Shape” panel, select “Quad” option.
2. From the “Type” panel, select “Mapped (four-sided areas)” option.
3. From the “Size” panel: (1) pick or type in “Line/s” edit box “1 4” and in “Divisions” edit box “100”, then press “Add” button; (2) pick or type in “Line/s” edit box “3 2” and in “Divisions” edit box “150”, then press “Add” button.

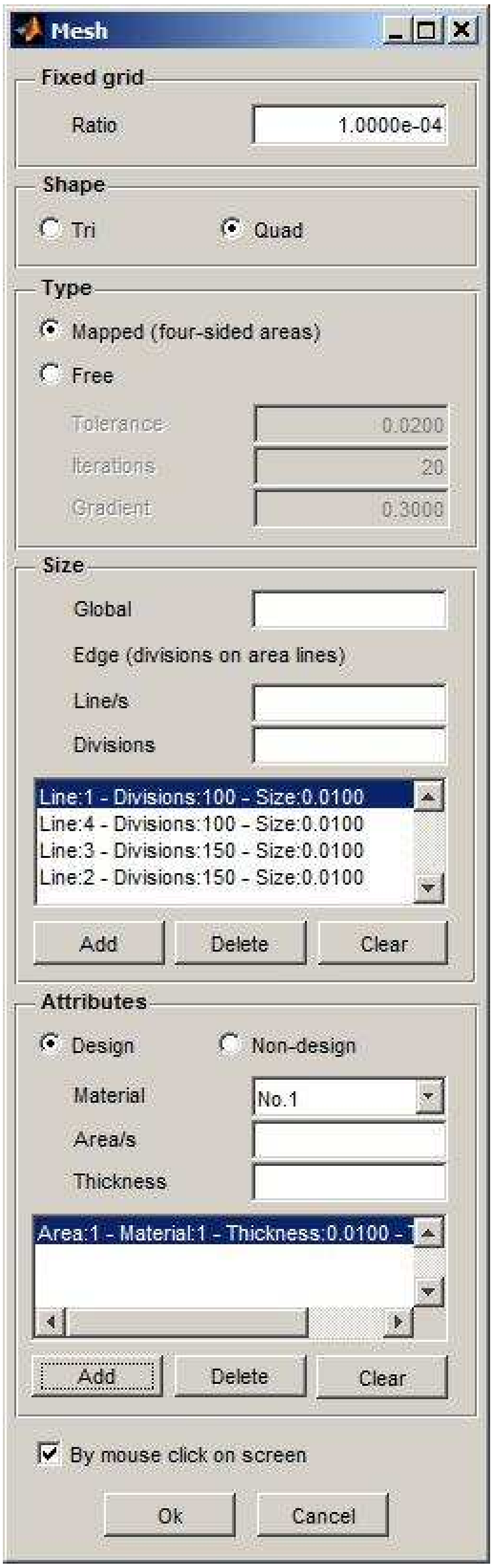

Figure 39. Mesh dialog box showing the mesh data.

4. From the “Attributes” panel, pick or type in “Area/s” edit box “1” and in “Thickness” edit box “0.01”, then press “Add” button. The user can review or manage the input data using the list boxes included in the dialog box (Figure 39).
5. Press “Ok” button. When the process is completed, the mesh of Figure 40 should be drawn.

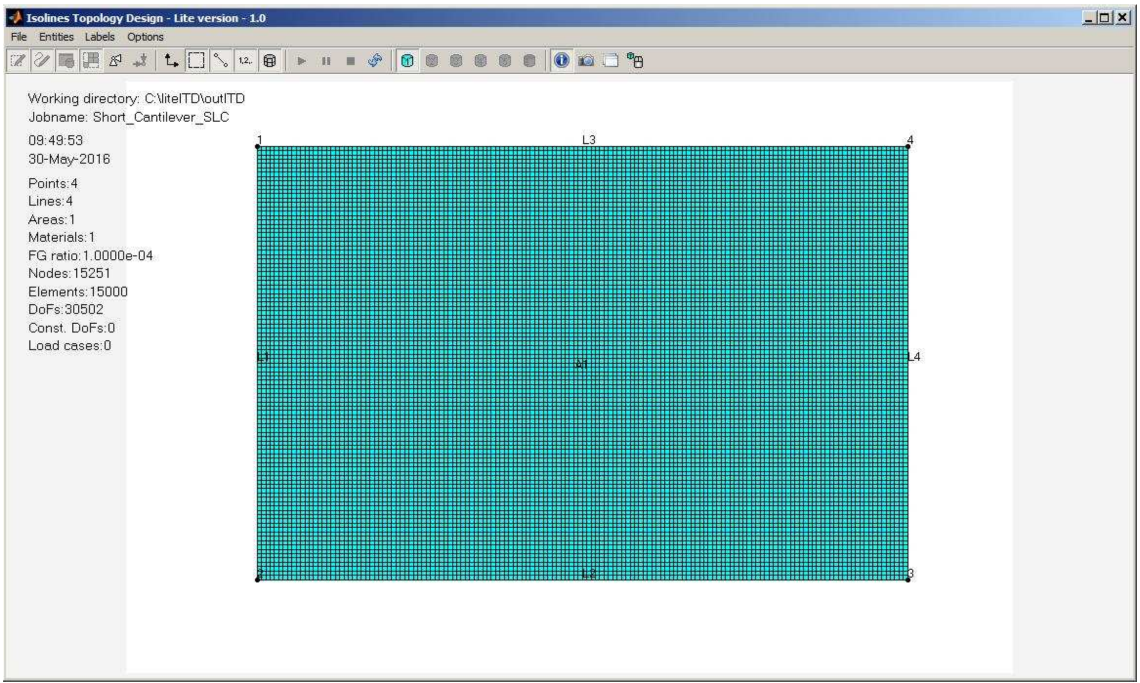

Figure 40. liteITD graphical interface showing the generated finite element mesh.

4.5 Applying DOF constraints

After the geometric model is meshed, the user can define DOF constraints at nodes. By clicking on the “Input support conditions” button, the “Displacements on nodes” dialog box will appear (Figure 25).

1. From “Selection” panel, select the “Box” radio button.
2. Move the mouse into the display area to upper left screen corner. At this location, pick the left mouse button, and move to the right and down until the first column of nodes is inside of the dashed blue box, then pick the left mouse button again. The selected nodes are marked using a blue square marker and their labels are shown in the “Node/s” edit box.

3. Press the “Add” button. The selected nodes will be included in the list box (Figure 41) and a two triangles (X- and Y- displacements are fixed) per constrained node will show in the display area (Figure 43).
4. When done, press the “Ok” button.

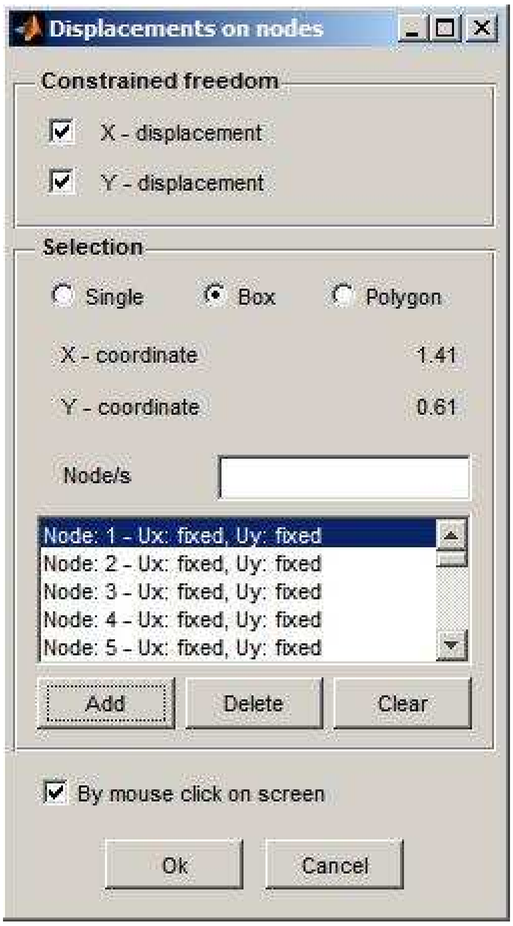

Figure 41. Displacements on nodes dialog box showing the DOF constraint data.

4.6 Applying loading conditions

The user can specify forces at nodes by clicking on the “Input nodal forces” button, the “Forces on nodes” dialog box will appear (Figure 26).

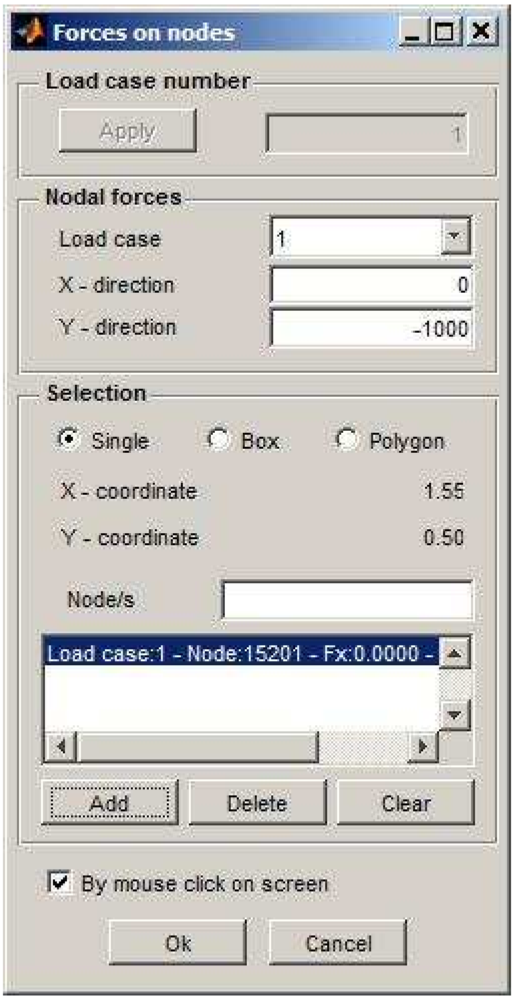

Figure 42. Forces on nodes dialog box showing the loading conditions data.

1. From the “Load case number” panel, press “Apply” button. For this example, the number of load cases need not be changed.
2. Input the force components as “0” for “X-direction” and “-1000” for “Y-direction”.
3. Move the mouse to the position (1.5, 0.5), the mouse position is displayed from the “Selection” panel. Press the left button to include the node nearest to the mouse location in the “Node/s” edit box.
4. Press “Add” button. The selected node will be included in the list box (Figure 42) and a red arrow pointing down will be drawn (Figure 43).

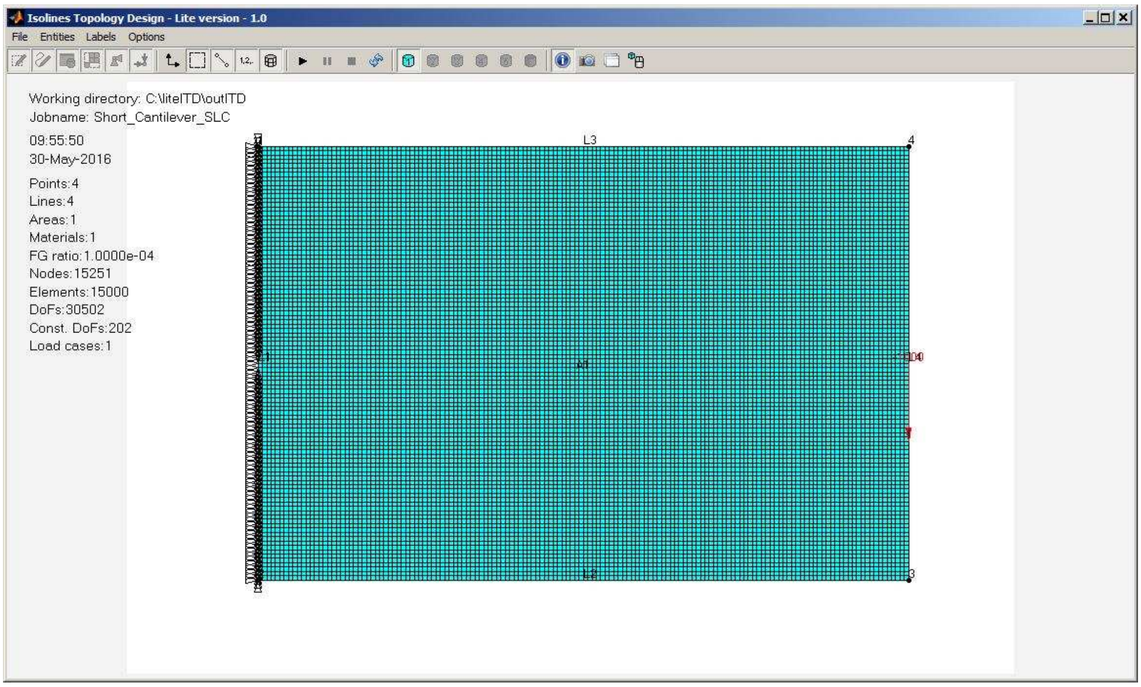

Figure 43. liteITD graphical interface showing the support and loading conditions.

4.7 Specify parameters and run liteITD

Press the “Optimization launcher” button. The dialog box of Figure 29 will appear. From the “Evolution” panel, replace “30” with “10” for “Obj. fraction (%)” edit box. Press the “Ok” button to accept these new parameters and to start the optimization.

4.8 View the resulting optimal design

After one hundred iterations, the optimization run is finished and the final design with 10% volume fraction will be obtained as shown in Figure 44.

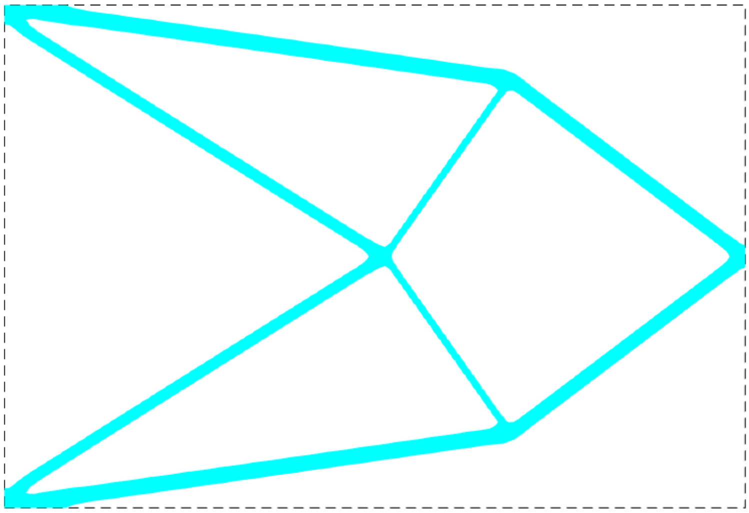

Figure 44. liteITD workbench showing the final optimal design for the short cantilever. 10% volume fraction, 100 iterations and 212 structural analysis.

The resulting topology with ITD algorithm (which resembles the layout of an 8-bar truss) reveals a good agreement with the optimum design obtained (Figure 45) using SIMP method (Sigmund, 2001).

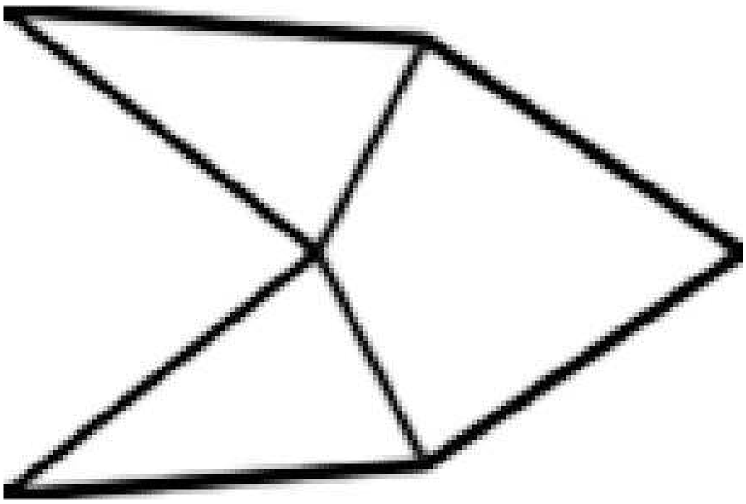

Figure 45. Final optimal design for the short cantilever using a 99 line topology optimization code written in Matlab (Sigmund, 2001) with the input line: top(150,100,0.1,3.0,1.5). 10% volume fraction, 199 iterations and 199 structural analysis.

5 ADDITIONAL EXAMPLES

In order to illustrate the other optimization algorithms implemented in the liteITD program, the short cantilever was optimized again: (1) under multiple loading conditions; (2) using multi-materials; (3) with different material properties in tension and compression.

5.1 Short cantilever under multiple loading conditions

Most real structures are subjected to different loads at different times (not all loads act simultaneously). This is referred to as multiple loading conditions (e.g. a moving load can be suitably simulated by multiple load cases). With multiple loading conditions, the resulting design has to be optimized to account for all load cases.

The geometric model, material properties, mesh characteristics, and support conditions used in this second example are the same as for the first example (see Section 4). The upward force F_1= 1000 N is applied at the upper right-hand corner and represents load case 1. The downward force F_2 = 1000 N is applied at the lower free end and represents load case 2, as shown in Figure 46. The ITD parameters were set as follows: total number of iterations 100; final volume fraction 20%; minimum volume change 1%.

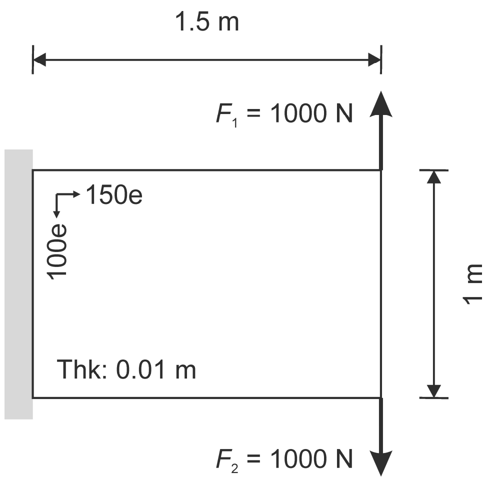

Figure 46. Design domain for the short cantilever under two load cases, where F_1 and F_2 represent each load case.

In order to carry out the topology optimization under multiple loading conditions, follow these steps:

1. Start liteITD.exe.
2. Open “Change jobname” dialog box, then type “Short_Cantilever_MLC” and press “OK”.

3. Repeat Sections 4.1 to 4.5 or load “Short_Cantilever_SLC_asv-sup.mat” file.
4. Press the “Input nodal forces” button. Type “2” in the “Load case number” edit box, then press “Apply” button.
5. From the “Nodal forces” panel, type “1000” in the “Y-direction” edit box.
6. Select the node located at the upper right-hand corner, then press “Add” button.
7. Select “2” from the “Load case” pop-up menu.
8. From the “Nodal forces” panel, type “-1000” in the “Y-direction” edit box.
9. Select the node located at the lower free end, then press “Add” button.
10. Press “Ok” button in order for these values to be applied.
11. Press the “Optimization launcher” button. From the “Load case” panel, select “Multiple” radio button.
12. From the “Evolution” panel, replace “30” with “20” for “Obj. fraction (%)” edit box. Press the “Ok” button to accept these new parameters and to start the optimization.

After one hundred iterations, the final optimal design under two load cases is shown in Figure 47.

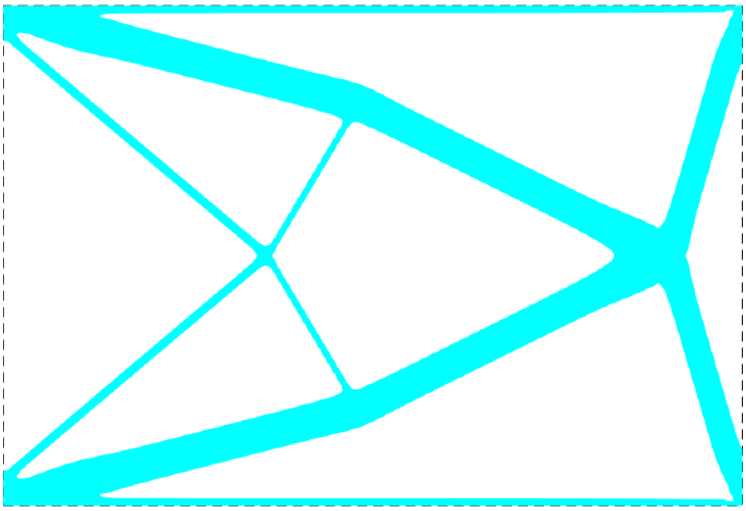

Figure 47. liteITD workbench showing the final optimal design under two load cases. 20% volume fraction, 100 iterations and 294 structural analysis.

Similar solution to this example (Figure 48) was obtained using SIMP method (Sigmund, 2001).

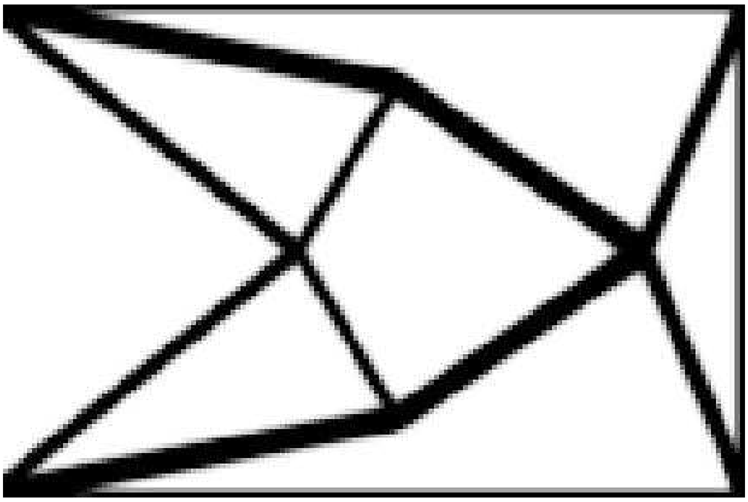

Figure 48. Final optimal design for the short cantilever under two load cases using a 99 line topology optimization code written in Matlab (Sigmund, 2001) with the input line: top(150,100,0.2,3.0,1.5). 20% volume fraction, 164 iterations and 328 structural analysis.

5.2 Short cantilever with different properties in tension and compression

The geometric model, material properties, mesh characteristics, support conditions, and forces applied are the same as for the first example, Figure 33. In this third example, we will study the use of the same material for tensile and compressive elements of the structure with different properties. Young's moduli of tensile and compressive members of the structure were 2×10^{11} Pa and 1×10^{11} Pa, respectively. Weight per unit volume was 78500 N/m^3. The ITD parameters were set as follows: total number of iterations 50; final volume fraction 25%; minimum volume change 1%.

In order to optimize the structure again using different properties in tension and compression, follow these steps:

1. Start liteITD.exe.
2. Open "Change jobname" dialog box, then type "Short_Cantilever_TC" and press "OK".
3. Repeat Sections 4.1 to 4.6 or load "Short_Cantilever_SLC_asv-for.mat" file.

4. Press the "Optimization launcher" button. From the "T-ension/C-ompression" panel, select "T/C–Material properties" radio button, then type "78500" for "T-density" and "C-density", "2E+11" for "T-Young's mod.", and "1E+11" for "C-Young's mod."
5. From the "Evolution" panel, replace "100" with "50" for "Iterations" edit box and "30" with "25" for "Obj. fraction (%)" edit box. Press the "Ok" button to accept these new parameters and to start the optimization.

The final optimal design is shown in Figure 49. The regions under tension/compression are displayed in red (lighter)/blue (darker) colours, respectively.

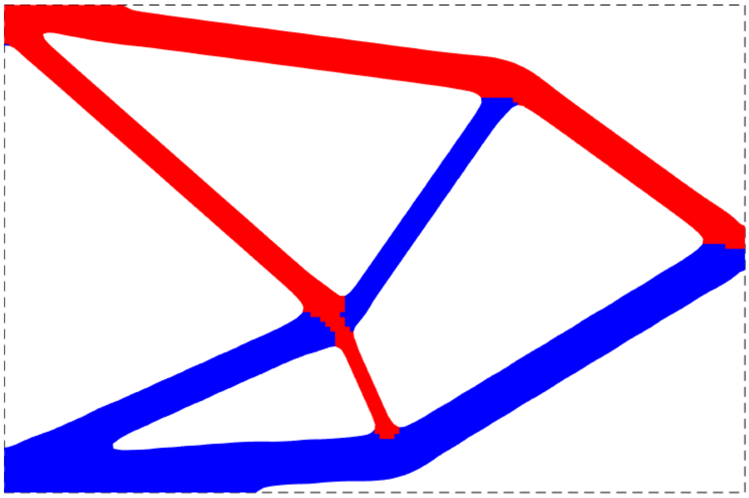

Figure 49. liteITD workbench showing the final optimal design with different material properties in tension and compression. 25% volume fraction, 50 iterations and 123 structural analysis.

The same problem was also optimized using the method of Martinez et al. (2007), producing the layout of Figure 50. The resulting topology with ITD algorithm is very similar to that of the truss-produced optimum.

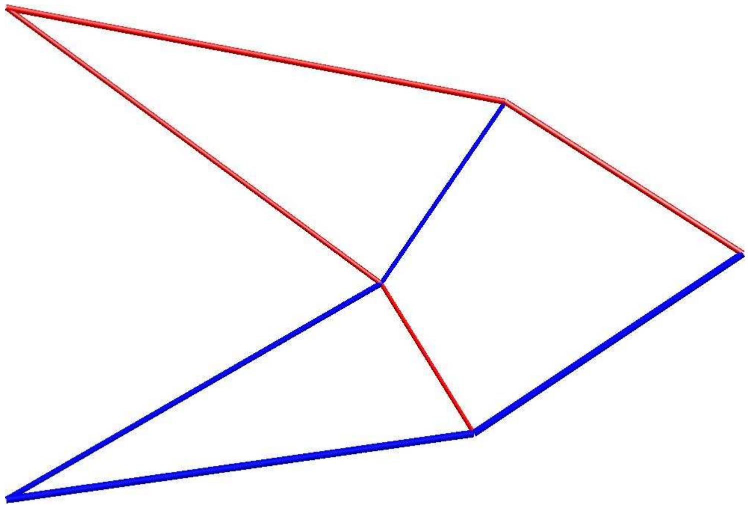

Figure 50. Optimal topology for the short cantilever using the method of Martinez et al. (2007) with 4 iterations and 72 structural analysis.

5.3 Short cantilever using multi-materials

The geometric model, mesh characteristics, support conditions, and forces applied are the same as for the first example, Figure 33. In this example, two materials were used, with Young's moduli of $E_1 = 2\times10^{11}$ Pa and $E_2 = 2\times10^{10}$ Pa, Poisson's ratios equal to 0.3, and volume control weighting factors equal to $w_1 = 0.5$ and $w_2 = 0.5$. The material with the highest stiffness (material No. 1) is displayed in magenta (darker colour), and the material No. 2 is displayed in cyan (lighter colour). The ITD parameters were set as follows: total number of iterations 50; final volume fraction 10%; minimum volume change 2%.

In order to optimize the structure again using two-material scheme, follow these steps:

1. Start liteITD.exe.
2. Open “Change jobname” dialog box, then type “Short_Cantilever_MM” and press “OK”.
3. Repeat Sections 4.1 to 4.2 or load “Short_Cantilever_SLC_asv-geo.mat” file.
4. Press the “Define material properties” button. From the “Number of materials” pop-up menu, select “2”. For the material No. 1, select “Magenta” from “Colour” pop-up

menu. For the material No. 2, select “Cyan” colour and type “2E+10” in the “Young’s modulus” edit box. Then Press “Ok”.

5. Repeat Section 4.4. In case of initial material distribution is unknown, the model can be meshed using only one material type.
6. Repeat Sections 4.5 to 4.6.
7. Press the “Optimization launcher” button. From the “Material” panel, select “Multiple materials (Fractions)” radio button. Click on the “Material 1” and “Material 2” check boxes, and then type “0.5” in both edit boxes.
8. From the “Evolution” panel, replace “100” with “50” for “Iterations” edit box, “1” with “2” for “Min. change (%)” edit box and“30” with “10” for “Obj. fraction (%)” edit box. Press the “Ok” button to accept these new parameters and to start the optimization.

The final optimal design for two-material scheme is shown in Figure 51. The obtained design is in good agreement with that obtained by Querin et al. (2015), Figure 52.

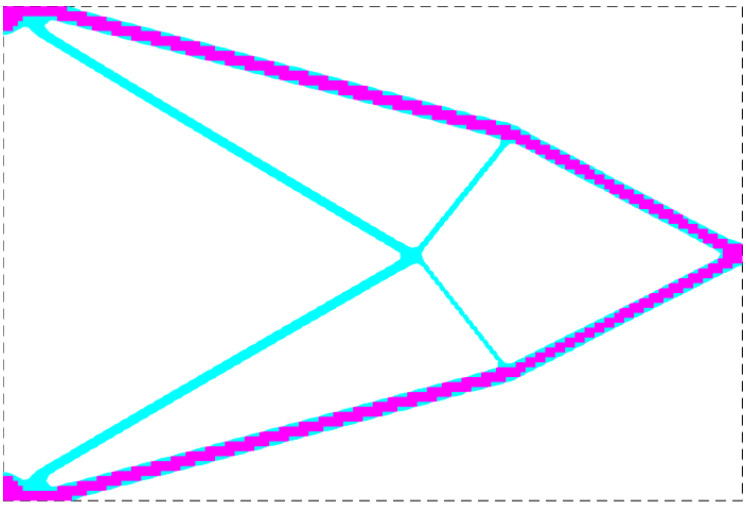

Figure 51. liteITD workbench showing the final optimal design for two-material scheme. 10% volume fraction, 50 iterations and 113 structural analysis.

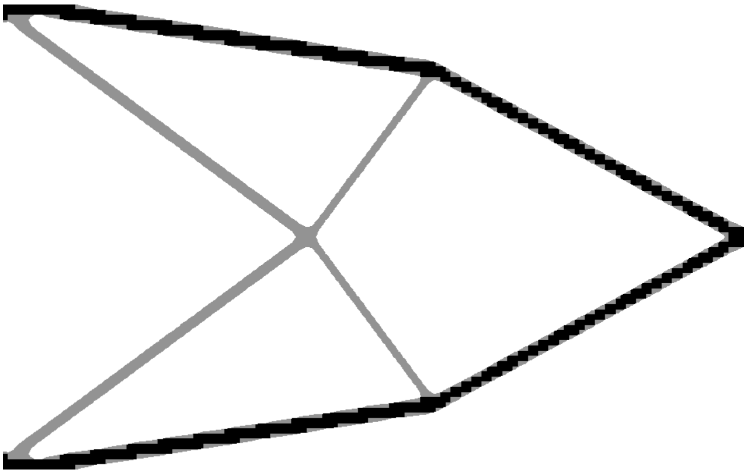

Figure 52. Final design for two-material scheme using the ITD method (Querin et al. 2015). 10% volume fraction, 50 iterations and 143 structural analysis.

6 CONCLUSIONS

In this paper, a software package called liteITD (lite version of Isolines Topology Design) for topology optimization of two-dimensional continuum structures was presented. This application was completely designed using MATLAB GUI environment. The usage of this application is directed to students mainly (educational purposes), although also to designers and engineers with experience. Several examples were presented to show the effectiveness of the program, which provides quality solutions with a very detailed contours without the need to interpret the topology in order to obtain a final design. liteITD can be downloaded and used for free from the website: http://www.upct.es/goe/software/liteITD.php.

ACKNOWLEDGEMENTS

This work has been partially supported by Spanish Ministry of Economy and Competitiveness", via the research project DPI2011-27939-C02-01. Its support is greatly appreciated.

REFERENCES

Aage N., Nobel-Jørgensen M., Andreasen C.S., Sigmund O. 2013. Interactive topology optimization on hand-held devices. Structural and Multidisciplinary Optimization. 47: 1-6.

Altair Engineering, Inc. Hyperworks. OptiStruct®. 2016.

Andreassen E., Clausen A., Schevenels M., Lazarov B.S., Sigmund O. 2011. Efficient topology optimization in matlab using 88 lines of code. Structural and Multidisciplinary Optimization. 43(1): 1-16.

ANSYS® Academic Research, Release 17. 2016.

Bendsøe M.P. 1989. Optimal shape design as a material distribution problem. Structural Optimization. 1: 193-202.

Bendsøe M.P. 1995. Optimization of Structural Topology, Shape and Material, Springer-Verlag.

Bendsøe M.P., Kikuchi K. 1988. Generating optimal topologies in structural design using a homogenization method. Computer Methods in Applied Mechanics and Engineering. 71: 197-224.

Bendsøe M.P., Sigmund O. 2003. Topology Optimization. Theory, Methods and Applications. Springer Verlag.

Challis V.J. 2010. A discrete level-set topology optimization code written in matlab. Structural and Multidisciplinary Optimization. 41(3): 453-464.

Coelho L.d.S., Sierakowski C.A. 2008. A software tool for teaching of particle swarm optimization fundamentals. Advances in Engineering Software. 39: 877-887.

Creative Engineering Services (CES). CATOPO®. 2016.

Cui C., Ohmori H., Sasaki M. 2003. Computational Morphogenesis of 3D Structures by Extended ESO Method. Journal of the International Association for Shell and Spatial Structures. 44(141): 51-61.

Dynamic Design Solutions (DDS). CAE software and services. NV. FEMtools®. 2016.

Garcia M.J., Henao M., Ruiz O.E. 2005. Fixed grid finite element analysis for 3D structural problems. International Journal of Computational Methods. 2(4): 569-585.

Garcia M.J., Ruiz O.E., Ruiz L.M., Querin O.M. 2004. Fixed grid finite element analysis for 3D linear elastic structures. In proceedings of the Computational Mechanics. WCCM VI in conjunction with APCOM'04, Beijing, China.

Garcia M.J., Steven G.P. 1999. Fixed grid finite elements in elasticity problems. Engineering Computations. 16(2): 145-164.

Garcia M.J., Steven G.P. 2000. Fixed grid finite element analysis in structural design and optimization. In proceedings of the Second ISSMO/AIAA Internet Conference on Approximations and Fast Reanalysis in Engineering Optimization.

Hinton A., Illingworth J. 1997. Marching Triangles: Delaunay implicit surface triangulation. Technical Report CVSSP 01, University of Surrey, Guildford, UK.

Koguchi A., Kikuchi N. 2006. A Surface Reconstruction Algorithm for Topology Optimization. Engineering with Computers. 20: 1-10.

Liu K., Tovar A. 2014. An efficient 3D topology optimization code written in MATLAB. Structural and Multidisciplinary Optimization. 50: 1175-1196.

Maan F.S., Querin O.M., Barton D.C. 2007. Extension of the fixed grid finite element method to eigenvalue problems. Advances in Engineering Software. 38: 607-617.

Martinez P., Martí P., Querin O.M. 2007. Growth method for size, topology, and geometry optimization of truss structures. Structural and Multidisciplinary Optimization. 33(1): 13-26.

MathWorks, Inc., 2016. Natick, MA, MATLAB R2016a®.

Michell A.G.M. 1904. The limits of economy of material in frame-structures. Philosophical Magazine. 8: 589-97.

MSC Software Corporation. MSC Nastran®. 2015.

Osher S., Sethian J.A. 1988. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. Journal of Computational Physics. 79: 12-49.

Paulino G.H., Page R.C., Nelli Silva E.C. 2005. A Java-based topology optimization program with web access: nodal design variable approach. 6th World Congresses of Structural and Multidisciplinary Optimization, Rio de Janeiro, Brazil.

Querin O.M. 1997. ESO Computer Program Evolve97. Chapter 9, Evolutionary Structural Optimization, Berlin, Heidelberg, New York: Springer.

Querin O.M., Steven G.P., Xie Y.M. 1998. Evolutionary structural optimization (ESO) using a bidirectional algorithm. Engineering Computations. 15: 1031-1048.

Querin O.M., Victoria M., Diaz C., Martí P. 2015. Layout optimization of multi-material continuum structures with the isolines topology design method. Engineering Optimization. 47(2): 221-237.

Querin O.M., Victoria M., Martí P. 2010. Topology optimization of truss-like continua with different material properties in tension and compression. Structural and Multidisciplinary Optimization. 42: 25-32.

Rozvany G.I.N., Lewiński T. 2014. Topology Optimization in Structural and Continuum Mechanics. CISM International Centre for Mechanical Sciences. Courses and Lectures Vol. 549. Springer Wien Heidelberg New York Dordrecht London.

Rozvany G.I.N., Zhou M., Birker T. 1992. Generalized shape optimization without homogenization. Structural Optimization. 4: 250-254.

Sigmund O. 2001. A 99 line topology optimization code written in MATLAB. Structural and Multidisciplinary Optimization. 21: 120-127.

Suresh K. 2010. A 199-line matlab code for Pareto-optimal tracing in topology optimization. Structural and Multidisciplinary Optimization. 42: 665-679.

Tcherniak D., Sigmund O. 2001. A web-based topology optimization program. Structural and Multidisciplinary Optimization. 22: 179-187.

TopOpt research group. DTU Mechanical Engineering & DTU Mathematics. TopOpt 2009.

Victoria M., Marti P., Querin O.M. 2009. Topology design of two-dimensional continuum structures using isolines. Computers & Structures. 87:101-109.

Victoria M., Querin O.M., Diaz C., Martí P. 2014. The effects of membrane thickness and asymmetry in the topology optimization of stiffeners for thin-shell structures. Engineering Optimization. 46(7): 880-894.

Victoria M., Querin O.M., Martí P. 2010. Topology design for multiple loading conditions of continuum structures using isolines and isosurfaces. Finite Elements in Analysis and Design. 46: 229-237.

Victoria M., Querin O.M., Martí P. 2011. Generation of strut-and-tie models by topology design using different material properties in tension and compression. Structural and Multidisciplinary Optimization. 44: 247-258

Victoria M., Querin O.M., Martí P. 2011. Topology design of three-dimensional continuum structures using isosurfaces. Advances in Engineering Software. 42: 671-679.

Wang M.Y., Chen S., Xia Q. 2004. Structural topology optimization with the level set method. http://www2.acae.cuhk.edu.hk/cmdl/download.htm.

Woon S.Y., Tong L., Querin O.M., Steven G.P. 2003. Optimising Topologies Through a Multi-GA System. In Proceedings of the Fifth World Congress of Structural and Multidisciplinary Optimization (WCSMO 5), Lido di Jesolo, 229-230.

Xie Y.M., Steven G.P. 1997. Evolutionary Structural Optimization, Berlin, Heidelberg, New York: Springer.

Zuo Z.H. 2010. Computer Program BESO2D. Chapter 10, Evolutionary Topology Optimization of Continuum Structures: Methods and Applications. John Wiley & Sons, Ltd.