Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

solveTruss v1.0: Static, global buckling and frequency analysis of 2D and 3D trusses with Mathematica

Hakan Ozbasaran

Eskisehir Osmangazi University, Department of Civil Engineering, Eskisehir, Turkey

ARTICLE INFO

Article history:
Received 31 January 2017
Received in revised form 16 May 2017
Accepted 17 May 2017

Keywords:
Truss
Buckling
Frequency
Mathematica

ABSTRACT

Trusses have an important place amongst engineering structures due to many advantages such as high structural efficiency, fast assembly and easy maintenance. Iterative truss design procedures, which require analysis of a large number of candidate structural systems such as size, shape and topology optimization with stochastic methods, mostly lead the engineer to establish a link between the development platform and external structural analysis software. By increasing number of structural analyses, this (probably slow-response) link may climb to the top of the list of performance issues. This paper introduces a software for static, global member buckling and frequency analysis of 2D and 3D trusses to overcome this problem for Mathematica users.

Code metadata

Current code version	v1.0
Permanent link to code/repository used of this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-17-00010
Legal Code License	GNU general public license (http://www.gnu.org/copyleft/gpl.html)
Code versioning system used	None
Software code languages, tools, and services used	Wolfram Mathematica
Compilation requirements, operating environments & dependencies	Wolfram Mathematica
If available Link to developer documentation/manual	None
Support email for questions	ozbasaran@ogu.edu.tr

1. Motivation and significance

Trusses are useful structures formed by connecting individual straight bar members at joints. These structural systems are loaded from their joints and their response to acting loads emerge as axial loads on the members. This favorable internal force distribution allows one to construct long-span and light-weight structures such as bridges, roofs, towers and cranes. Since members of the trusses only carry axial loads, only two conditions may apply such as tension and compression. Therefore, analysis of truss structures and dimensioning of their members are relatively simple compared to other structural systems. In spite of their simple behavior, it is not easy to design a structure with high benefit/cost ratio which satisfies all of the requirements. In practice, engineers generally use commercial finite element analysis (FEA) software packages such as SAP2000 [1], ABAQUS [2] and ANSYS [3] for analysis and design of structural systems. These software can perform static, dynamic and buckling analyses of complex structure models including trusses. Some of these software packages provide scripting and application programming interfaces (API) compatible with various programming languages.

Optimal design of trusses has become one of the most competitive fields in structural engineering research in the last decades. Optimization of trusses deals with searching for the optimal sections (size optimization), node coordinates (shape optimization) and element information (topology optimization) while satisfying the design requirements. These design requirements, which are called “constraints”, may be limiting stresses, deflections, frequencies etc. for the individual structural elements or the whole structure. Since optimization algorithms may include advanced mathematical operations, most researchers direct their attention towards computer algebra systems which provide a development platform and a built-in mathematical functions library.

E-mail address: ozbasaran@ogu.edu.tr.

http://dx.doi.org/10.1016/j.softx.2017.05.004

Truss optimization is generally an iterative process that requires repetitive structural analysis of the candidate designs and check of the constraint violations. Thousands of candidate designs must be analyzed to obtain a near-optimal solution with most of the stochastic optimization algorithms [4–13]. One second delay in the analysis of one truss system causes a procedure of 10 000 truss analyses to last 2.78 hours longer. There are two main choices to perform the structural analyses. The first one is to establish a link between the development platform and a structural analysis software. Of course, it is necessary that the structural analysis software has an interface for this purpose or it is able to read a text input file which will be generated by the algorithm. The second one is building a structural analysis procedure on the preferred platform and calling it when needed which mostly provides faster calculation.

The main purpose of the presented software is to perform static, global member buckling and frequency analysis of 2D and 3D trusses within the Mathematica [14] development platform (instead of using external software) to boost the performance of iterative truss analysis and design processes for Mathematica users. The only package for Mathematica that the author can find in the literature is developed by Zamiatina [15] in 1999 for static analysis of structures. solveTruss calculates joint deflections, internal forces, elastic critical global buckling loads and demand/capacity ratios of the members. It determines possible failure modes of the members and natural periods of the structure. It also presents useful information such as total volume of the material used and total penalty of the design under a given loading case. Following part describes the software and the input format.

2. Software description and input format

The solveTruss software uses the Mathematica platform, which is a widely used computer algebra system. It is constructed as procedures in order to provide an easy-to-implement code. "solveTruss2D" is for plane trusses and "solveTruss3D" is for spatial trusses of which parameters are the same as follows;

$$\begin{aligned} & solveTruss2D[\{E,\nu,\sigma_a,\delta_a,\lambda_a\}, joints, sections, bars, \\ & \quad supports, forces, masses, mode] \qquad 2D \\ & solveTruss3D[\{E,\nu,\sigma_a,\delta_a,\lambda_a\}, joints, sections, bars, \\ & \quad supports, forces, masses, mode] \qquad 3D \end{aligned} \tag{1}$$

where, E is Young's modulus, ν is Poisson's ratio, σ_a is allowed absolute stress, δ_a is allowed absolute deflection for joints in global directions and λ_a is allowed eigenvalue for buckling analysis. The rest are the lists which describe the structure geometry, loading conditions and requested natural periods. The coordinates of the joints are defined by the "*joints*" variable as given below;

$$\begin{aligned} & joints = \{\{joint\ ID, \dot{x}, \dot{y}\}, \ldots, \ldots\} \qquad 2D \\ & joints = \{\{joint\ ID, \dot{x}, \dot{y}, \dot{z}\}, \ldots, \ldots\} \qquad 3D. \end{aligned} \tag{2}$$

It can be seen from the Eq. (2) that the structure of the "*joints*" list varies due to if the system is planar or spatial. The same notation is used in the further text for 2D and 3D procedures. *joint ID* is the joint label. $\dot{x}$, $\dot{y}$ and $\dot{z}$ are the Cartesian coordinates of the joint with respect to the global coordinate system. Origin and orientation of the global coordinate system can be freely set by the user. The "*sections*" variable defines the sections and their properties (Eq. (3)).

$$sections = \left\{\left\{section\ ID, A, I_x, I_y, I_t, I_w, \{x_0, y_0\}\right\}, \ldots, \ldots\right\} \tag{3}$$

where, *section ID* is the section label, A is the section area, I_x and I_y are the moments of inertia about x and y principal centroidal axes of the section, respectively. I_t is the torsional constant, I_w is the warping constant and finally, x_0 and y_0 are the distances between center of mass (GC) and shear center (SC) in the x and y directions (Fig. 2.1). Note that the moments of inertia about the principal centroidal axes, torsional and warping constants and relative shear center location information are requested in addition to section area. Hence, these properties are required for global buckling analysis of the bars (see [16–18] for details).

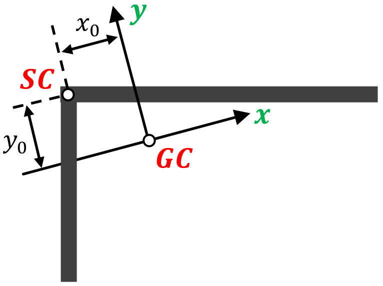

Fig. 2.1. An unsymmetrical cross section (GC: center of mass, SC: shear center).

The variable "*bars*" provides the element and connectivity information. In truss structures each element connects a joint to another. These joints are represented by *start joint ID* and *end joint ID* in the "*bars*" list. *section ID* assigns the corresponding section to the member labeled with *bar ID* (see Eq. (4)).

$$\begin{aligned} & bars \\ & \quad = \{\{bar\ ID, section\ ID, start\ joint\ ID, end\ joint\ ID\}, \ldots, \ldots\}. \end{aligned} \tag{4}$$

The variable "*supports*" is for the restraint conditions as given in Eq. (5).

$$\begin{aligned} & supports = \left\{\left\{joint\ ID, res_{\dot{x}}, res_{\dot{y}}\right\}, \ldots, \ldots\right\} \qquad 2D \\ & supports = \left\{\left\{joint\ ID, res_{\dot{x}}, res_{\dot{y}}, res_{\dot{z}}\right\}, \ldots, \ldots\right\} \quad 3D. \end{aligned} \tag{5}$$

In Eq. (5), $res_{\dot{x}}$, $res_{\dot{y}}$ and $res_{\dot{z}}$ are the restraint conditions for displacements in the $\dot{x}$, $\dot{y}$ and $\dot{z}$ directions, respectively which take the value 1 for "restrained" and 0 for "free" cases. The variable "*forces*" contains external force data of which structure is similar to "*supports*" list (Eq. (6)).

$$\begin{aligned} & forces = \left\{\left\{joint\ ID, F_{\dot{x}}, F_{\dot{y}}\right\}, \ldots, \ldots\right\} \qquad 2D \\ & forces = \left\{\left\{joint\ ID, F_{\dot{x}}, F_{\dot{y}}, F_{\dot{z}}\right\}, \ldots, \ldots\right\} \quad 3D \end{aligned} \tag{6}$$

where, $F_{\dot{x}}$, $F_{\dot{y}}$ and $F_{\dot{z}}$ are the forces acting in the $\dot{x}$, $\dot{y}$ and $\dot{z}$ directions, respectively. solveTruss does not convert the forces to masses. Masses should be entered separately in the variable "*masses*" to be considered in the frequency analysis (Eq. (7)).

$$\begin{aligned} & masses = \left\{\left\{joint\ ID, m_{\dot{x}}, m_{\dot{y}}\right\}, \ldots, \ldots\right\} \qquad 2D \\ & masses = \left\{\left\{joint\ ID, m_{\dot{x}}, m_{\dot{y}}, m_{\dot{z}}\right\}, \ldots, \ldots\right\} \quad 3D. \end{aligned} \tag{7}$$

Finally, the variable "*mode*" provides the number of the requested natural periods (r_m), lower (T_A) and upper limit (T_B) of the allowed natural period region (Eq. (8)). If r_m is set to 0, frequency analysis will not be performed.

$$mode = \{r_m, T_A, T_B\}. \tag{8}$$

It should be noted that it is not necessary to provide the lists ordered by labels (*joint ID*, *section ID* etc.). However, labels should be given starting from 1 and without skipping any number.

3. Structure and output

solveTruss uses the direct stiffness method to calculate joint deflections, which is very appropriate for programming. After the