5-2012

A Validation and Comparison About VABS-IDE and VABS-GUI

Teng Hu
Utah State University

A VALIDATION AND COMPARISON ABOUT VABS-IDE AND VABS-GUI

by

Teng Hu

A report submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Mechanical Engineering

Approved:

Dr. Wenbin Yu Major Professor	Dr. Ling Liu Committee Member
Dr. Thomas H. Fronk Committee Member	

UTAH STATE UNIVERSITY
Logan, Utah

2012

Abstract

A Validation and Comparison of VABS-IDE and VABS-GUI

by

Teng Hu, Master of Science

Utah State University, 2012

Major Professor: Dr. Wenbin Yu
Department: Mechanical and Aerospace Engineering

Variational Asymptotic Beam Section (VABS) analysis is a computer code for design and analysis of slender structure such as rotor blades, wind turbine blades, missiles, and rockets more efficiently than a full three-dimensional finite element analysis. VABS enables a rigorously a dimensional reduction of the original three-dimensional nonlinear analysis into a one-dimensional nonlinear beam analysis and a two-dimensional linear cross-sectional analysis. The VABS code has been extensively used in academia and industry for computing the sectional properties and recovering three-dimensional fields of composite slender structures. Recently two graphic user interface, namely VABS-IDE and VABS-GUI, have been developed to automatic the preprocessing and postprocessing for the VABS software. This report uses a few benchmark examples, including a circular aluminum tube, a highly heterogeneous section, a multi-layer composite pipe, and an isotropic blade-like section to validate and compare both tools. And also this report studies the mesh convergence of VABS and loss of accuracy using smeared properties, two questions raised by an industry user.

(64 pages)

Contents

List of Tables

List of Figures

Acronyms

VABS	Variational Asymptotic Beam Section analysis
PreVABS	A design driven matlab code for creating VABS inputs for Blade/Wing Sections
VABS-IDE	VABS Integrated Design Environment, developed by Advanced Dynamics Inc.
VABS-GUI	VABS Graphic User Interface, developed by Utah State University

Chapter 1

Introduction

To meet the increasing performance requirement of engineering structures, more and more structures are made of composite materials. Many load-bearing components, such as helicopter rotor blades and wind turbine blades, are slender structures having one dimension much larger than the other two dimensions. With the advances of computer hardware and software in recent years, the behavior of such structures can be accurately predicted using three-dimensional (3D) finite element analysis (FEA) such as those available in commercial packages ANSYS, ABAQUS, or NASTRAN, with a fine enough mesh. However, 3D FEA are too time-consuming to be used for effective design and analysis.

Common engineering approach to design and analyze slender structures is to use the beam theory. A beam theory requires a set of beam stiffness along with a closed system of the beam equations governing the behavior of beam reference line. For beams made of isotropic homogeneous materials, the beam constants such as extension stiffness, bending stiffness can be computed by some analytical integrals over the cross-sectional domain. The computation of torsional stiffness and transverse shear stiffness usually require the solution of partial differential equations over the cross-sectional domain which can solved exactly only for some very simple cross-sections. For general cross-sections, either some assumptions must be made for simple analytical solutions or numerical techniques should be used to solve these equations. When the slender structures are made of composite materials and featuring arbitrary external and internal geometry, numerical techniques must be used to achieve an accurate modeling of such structures.

1.1 VABS theory and code

The computer program VABS (Variational Asymptotic Beam Section Analysis) uses the variational asymptotic method to split a three-dimensional (3D) nonlinear elasticity problem into a two-dimensional (2D), linear, cross-sectional analysis and a one-dimensional (1D), nonlinear beam analysis. This is accomplished by taking advantage of certain small parameters inherent to beam-like structures. VABS is able to calculate the cross-sectional properties and 3D field distribution over the cross-section, with transverse shear and Vlasov refinements, for initially twisted and curved beams with arbitrary geometry and material properties [2]. The efficiency and accuracy of VABS have been comprehensively demonstrated and validated by many examples in its various publications [3].

Although VABS can compute the stiffness matrix for various common engineering beam models, the focus of this report is on the generalized Timoshenko beam model which can be described by a 6 × 6 stiffness matrix as follows:

$$\begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ M_1 \\ M_2 \\ M_3 \end{Bmatrix} = \begin{bmatrix} S_{11} & S_{12} & S_{13} & S_{14} & S_{15} & S_{16} \\ S_{12} & S_{22} & S_{23} & S_{24} & S_{25} & S_{26} \\ S_{13} & S_{23} & S_{33} & S_{34} & S_{35} & S_{36} \\ S_{14} & S_{24} & S_{34} & S_{44} & S_{45} & S_{46} \\ S_{15} & S_{25} & S_{35} & S_{45} & S_{55} & S_{56} \\ S_{16} & S_{26} & S_{36} & S_{46} & S_{56} & S_{66} \end{bmatrix} \begin{Bmatrix} \gamma_{11} \\ 2\gamma_{12} \\ 2\gamma_{13} \\ \kappa_1 \\ \kappa_2 \\ \kappa_3 \end{Bmatrix} \tag{1.1}$$

where F_1 is the extension force, F_2, F_3 are the transverse shear forces, M_1 is the torque, M_2, M_3 are the bending moments, γ_{11} is the tensile strain, $2\gamma_{12}$ and $2\gamma_{13}$ are the engineering transverse shear strains, κ_1 is the twist rate, and κ_2, κ_3 are the bending curvature. Here we choose coordinate x_1 to be the beam reference line, coordinate x_2 and x_3 to describe the cross-sectional domain. The stiffness matrix for a composite beam is in general a fully populated stiffness matrix The diagonal term S_{11} is commonly called extension stiffness and denoted using EA, S_{22}, S_{33} are commonly called the transverse shear stiffnesses, S_{44} is commonly called torsional stiffness and denoted using GJ, S_{55} is commonly called bending stiffness around x_2 direction and denoted using EI_{22}, S_{66} is commonly called bending stiffness around x_3 direction and denoted using EI_{33}.

VABS is a 2D analysis based on the finite element method (FEM) with a typical element library (triangular elements with 3-6 nodes and quadrilateral elements with 4-9 nodes). It is fully modularized and can be easily integrated into any CAD/CAM software. The code is of commercial quality and it has no problem to analyze realistic structures such as composite rotor blades with hundreds of layers on a personal computer within seconds. Although the nodal coordinates and elemental connectivity are compatible with formats used in commercial finite element analysis (FEA) codes, VABS was designed to for the cross-sectional analysis of composite slender structures. Some special information such as layer plane angle is not readily available from the mesh generator of commercial FEA codes and this was the major bottleneck for using VABS in an industry setting; for the layer plane angle, which not mean the lay-up angle; taking an example with a box beam, if there is a Cartesian coordinates system with x, y, and z direction, the lay-up angle on the plane that perpendicular to the x direction and the lay-up angle, which is perpendicular to the z direction is different (0 degree and 90 degree); to this end, VABS cannot handle with these two different angles; therefore, we said layer plane angle in not readily available. Therefore, some special purpose pre-processors and post-processors, including PreVABS, VABS-IDE, and VABS-GUI, are motivated to facilitate the preparation of VABS input files.

1.2 PreVABS

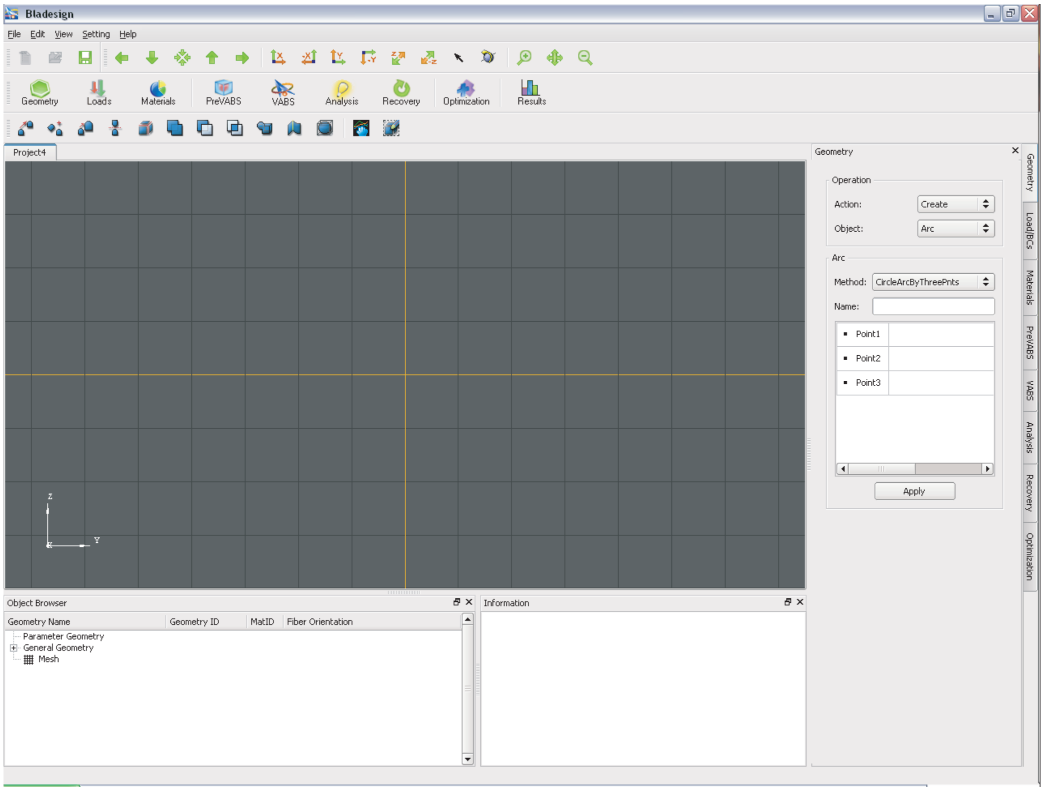
PreVABS is a design driven pre-processing computer program which can effectively generate high-resolution finite element modeling data for VABS by directly using design parameters such as simple geometric parameters and both the spanwisely and chordwisely varying composite laminate lay-up schema for rotor blade and aircraft wing sections. It was developed at Utah State University (USU) with the major funding provided by Groen Brothers Aviation. It has the capability of modeling sophisticated cross-sectional configurations for various composite helicopter rotor blades, wind turbine rotor blades, and aircraft wing sections. Most importantly, it has the merit of dramatically reducing the intensive modeling efforts for generating 3D FEA model which is either time costly or impractically especially during the preliminary and intermediate design phases. The accuracy of the finite element models generated by PreVABS and results obtained by VABS have been extensively validated in Ref. [1]. The code has been frequently requested and purchased along with VABS and used as a popular preprocessor in helicopter and wind turbine industry. The advantage of PreVABS is that it can generate the finite element model for the cross-section using a few design parameters. However, it lacks a graphic user interface and its modeling capability is limited to regular rotor blade and wing sections with some restricted internal constructions.

1.3 VABS-IDE

VABS-IDE stands for VABS Integrated Design Environment and it was developed by Advanced Dynamics Incorporation in collaboration with USU and Georgia Institute of Technology through a US Army Small Business Innovation Researh (SBIR) grant (Phase I and Phase II). VABS-IDE is designed to be a user-friendly, efficient high-fidelity composite rotor blade and wing section design environment for rapid and confident aeromechanics assessment during conceptual design stages. The geometry user interface environment seamlessly integrates the Variational Asymptotic Beam Section (VABS) analysis, the best proven technology for realistic composite rotor blade design, with a versatile CAD environment, a mesh generator, a robust optimizer, and a general-purpose post-processor, all of which are specially tailored for blade and wing section design. VABS-IDE provides all the tools required to design and analyze rotor blades and wings, from geometry building to the final steps of visualization and post-processing. It enables mechanical designers to create parts with confidence and fidelity similar to utilizing a much more resource intensive 3D finite element analysis procedure. The starting interface of the software is shown in Fig. 1.1.

1.4 VABS-GUI

VABS-GUI stands for VABS Graphic User Interface and it was developed at Utah State University under the sponsorship of Technology Commercialization & Innovation Program (TCIP) from Utah Governor's Office of Economic Development. It is a general-purpose graphic user interface specially designed for VABS based on gmsh, a open-source, general-



Action: Create

Object: Arc

Method: CircleArcByThreePnts

Fig. 1.1: The Interface of VABS-IDE.

purpose, finite element preprocessor [4]. It does not have an optimization module in comparison to VABS-IDE. However, it has a versatile CAD and mesh capability as it leverages the preprocessing capability provided by gmsh and it is easily extendable to add other capabilities. The same VABS code is also used as the analysis engine in VABS-GUI. The starting interface of the software is shown in Fig. 1.2.

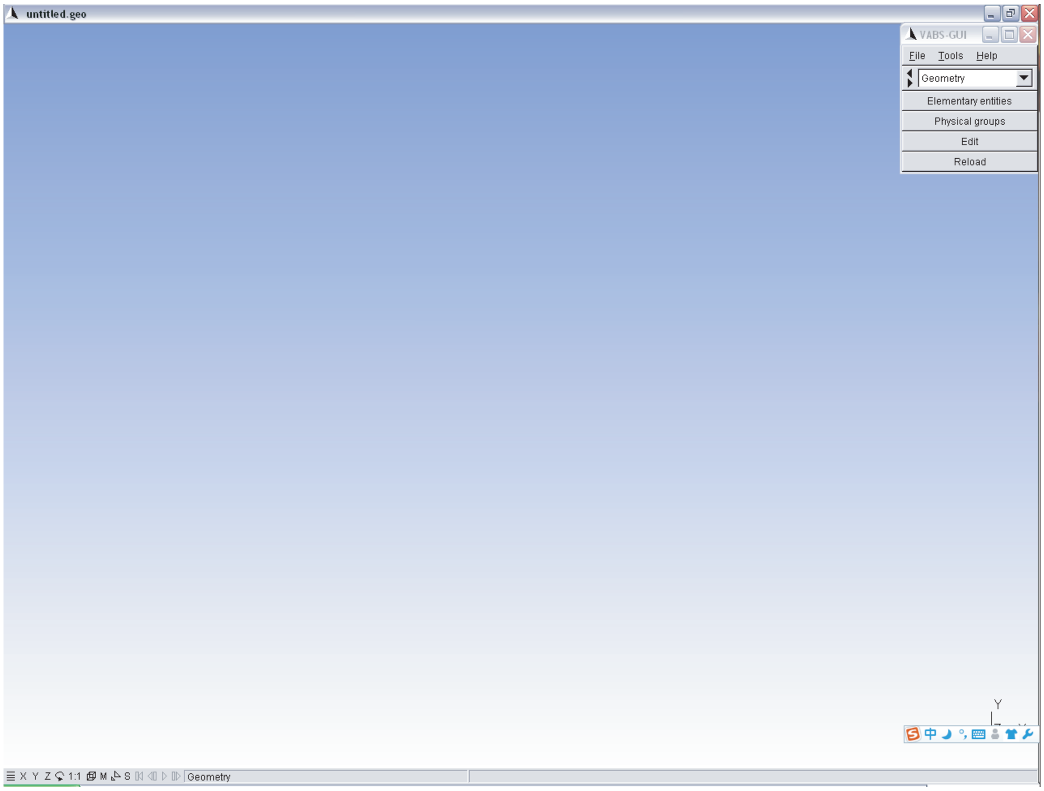


Fig. 1.2: The Interface of VABS-GUI.

The normal procedure of VABS-IDE and VABS-GUI needed for a cross-sectional analysis is:

1. Create the geometry using the CAD capability

2. Add material properties and possible lay-up information

3. Use mesh generator to create the finite element mesh

4. Use VABS for the analysis

From the first impression of these two tools, as shown in Fig. 1.1 and Fig. 1.2, one can see that there are differences between VABS-IDE and VABS-GUI. The author of this report personally feels the interface of VABS-GUI is very clean and tidy with a large geometry panel to do the modeling, and this advantages become clear in working out the examples in the following chapter. On the other hand, for VABS-IDE, the interface has many functional keys to help the user have a comprehensive understanding of how to use the input of geometry and material properties. However, the space left for the geometry panel is too small.

In comparison to PreVABS, both VABS-IDE and VABS-GUI have a versatile CAD modeling capability and a user friendly graphic user interface. However, a comprehensive demonstration and validation of both tools are not done yet. As we know that the accuracy of prediction not only depends on VABS, the same analysis engine for both VABS-IDE and VABS-GUI, but also the inputs prepared by pre-processors for VABS to carry out the analysis. Hence it is important to use some benchmark examples to validate whether VABS-IDE and VABS-GUI can correctly generate VABS inputs. In the mean time, it is useful to report the user experiences for these two software programs so that common VABS users can have an impression of both VABS-IDE and VABS-GUI and choose the right tool best suitable for their work dealing with design and analysis of composite beams.

1.5 Objectives of this report

The first objective of this report is to validate and compare both VABS-IDE and VABS-GUI as two preprocessors for VABS. Five benchmark examples including a circular aluminum tube, a highly heterogeneous section, a multi-layer composite pipe, and an isotropic blade-like section are used to achieve this objective. These examples are chosen because they are the examples used in Ref. [1] to validate PreVABS and we can compare our results those published and validated in that paper.

The second objective of this report is to study the mesh convergence of VABS as requested by an industry user using a simple example.

The third objective of this report is to investigate the loss of accuracy using the smeared property approach to model composite beams. The smeared property approach is used extensively in industry as most of the proprietary tools cannot provide a layer-by-layer modeling of composite beams. Although the smeared property approach is not necessary for VABS modeling, we can use VABS to quantify the loss of accuracy using this approach and the gain in simplicity and efficiency in modeling.

Chapter 2

Benchmark Examples

To validate and compare VABS-IDE and VABS-GUI, the first four benchmark examples used in Ref. [1] for validation of PreVABS will be reproduced using these two software programs.

2.1 A circular aluminum tube

The first example is a circular tube made of aluminium which has a radius of R=0.3 m with the Young's modulus of E=73 GPa, Poison's ratio of $\nu = 0.33$ and density of $\rho = 2800$ kg/m^3. With the origin at the center, only diagonal terms of the cross-sectional stiffness and mass matrix are not zero. Here we take the inner radius $t/2R = 1/3$.

It is straight forward to create the geometry using VABS-IDE as shown in Fig. 2.1. One just needs to create a center point and a radius, then click the “apply” button to create the circle. Then open the “object” panel to create a face. Here, VABS-IDE has a very convenient way to create a cross-section and the overall procedure takes only a couple of minutes.

On the other hand, it is a little bit more involved to create the geometry using VABS-GUI as shown in Fig. 2.2. In VABS-GUI, there is no command that make a circle directly by a center point and radius. A circle should be created in terms of four arcs. First, one should create a center point; then, create four more points both on the circle using its XY coordinates; the next step is to use the “arc” command to create four arcs using two points on the circle and the center point. Repeat this step four times to create a circle.

Both tools also are different in creating the finite element meshes for the cross-section. In VABS-IDE, after the geometry of the cross-section is created, the PreVABS panel can be used to generate the mesh according to the the element size, element type, and fineness

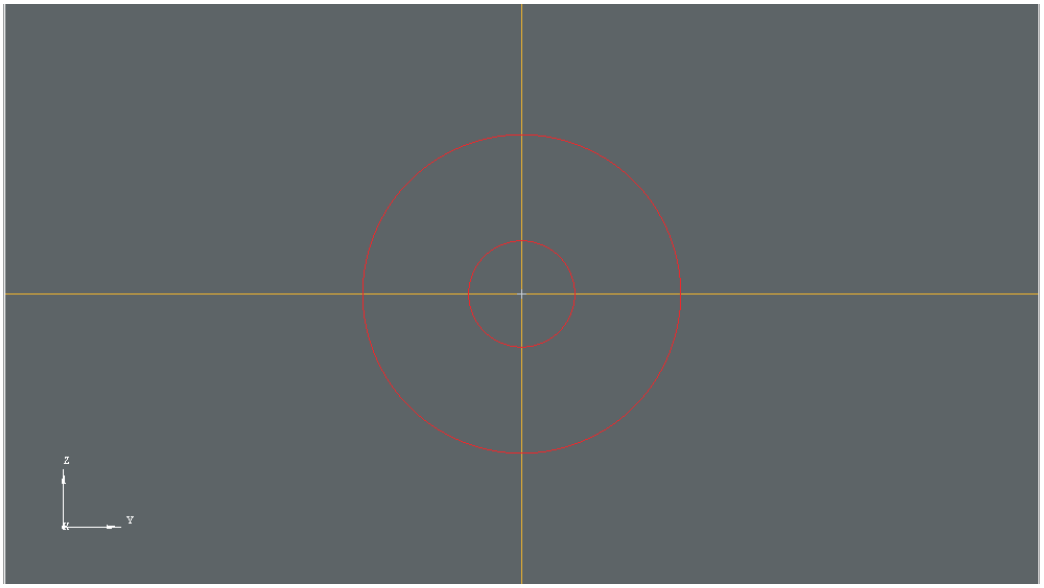


Fig. 2.1: Geometry of Circular Aluminium Tube (VABS-IDE)

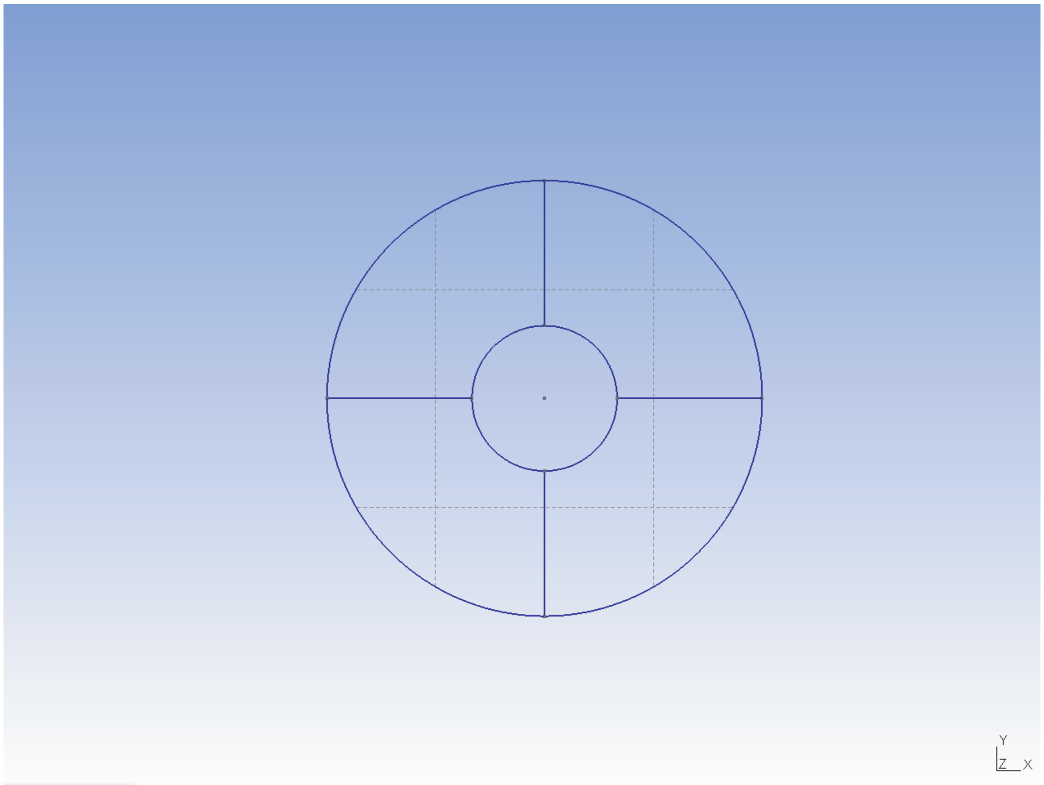


Fig. 2.2: Geometry of Circular Aluminium Tube (VABS-GUI)

Table 2.1: Mesh data for generating the first mesh of aluminum tube using VABS-IDE

Heal toler	Max size	Min size	Fineness	Element type	# of Elements
0.01	0.045	0.018	User Defined	TRI3	159

chosen by the user. Fig. 2.3 shows a mesh created by VABS-IDE. This mesh has a total of 159 elements and the input parameters for generating this mesh is listed in Table 2.1.

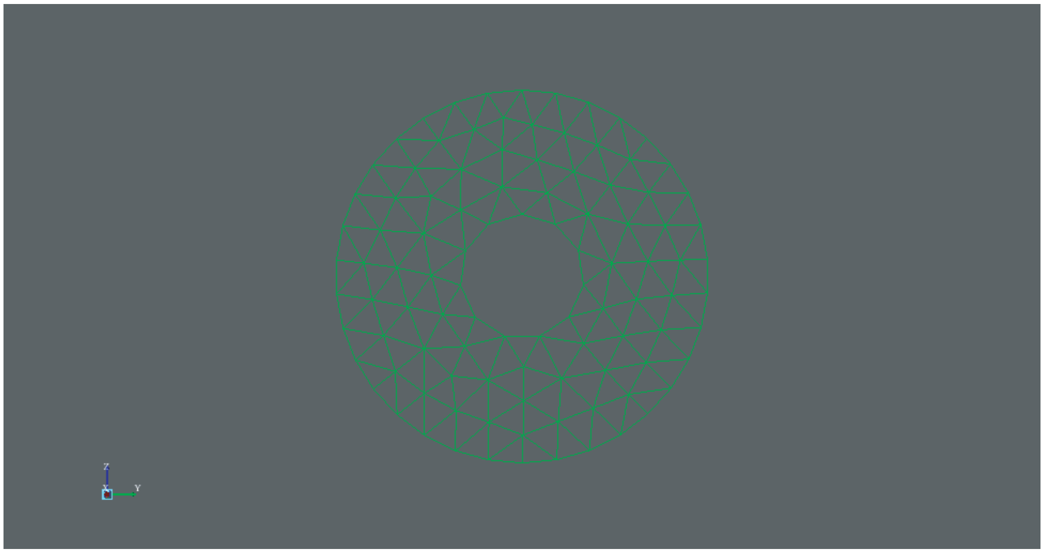


Fig. 2.3: First Mesh for Circular Aluminium Tube (159 elements, VABS-IDE)

Unlike the geometry creation, it is very convenient for VABS-GUI to create the finite element mesh. One just needs to first click “2D”, which means this cross-section will be meshed using 2D elements; then click “Refine by splitting” as needed to create a desired mesh. In order to compare and validate the results generated by VABS-IDE and VABS-GUI, we should keep the total number of elements comparable as it is well known that the accuracy of a finite element analysis depends on the number of elements we used in the finite element mode. Therefore, the total elements generated by VABS-GUI is controlled to be close to the total number of VABS-IDE of the same example. Since the corresponding mesh generator in these two codes are different, it is very difficult to obtain the same total number for elements. For this cross-section, after clicking “Refine by splitting” two twice in VABS-GUI, a mesh of 161 elements are generated as shown in Fig. 2.4.

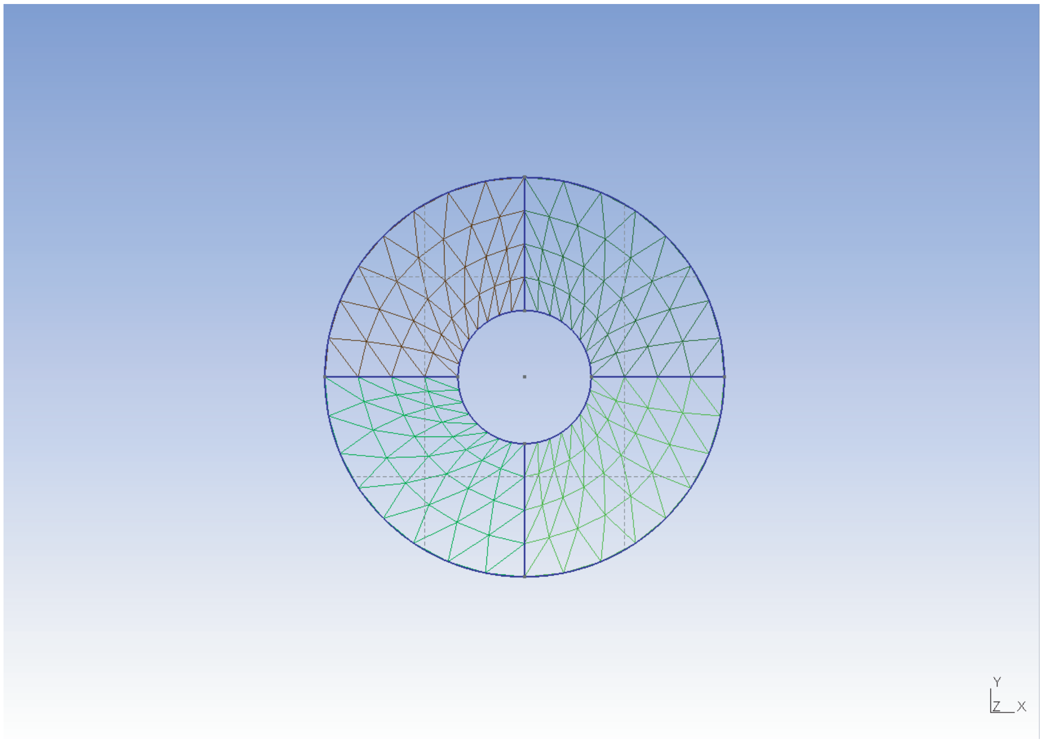


Fig. 2.4: First Mesh for Circular Aluminium Tube (161 elements, VABS-GUI)

Then one can use the VABS analysis code embedded in both software programs to perform the cross-sectional analysis and the 6×6 Timoshenko stiffness matrix results output by VABS-IDE are listed as follows:

$$\begin{bmatrix} 1.835\times10^{10} & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ & 4.835\times10^{9} & 1.574\times10^{5} & 0.0 & 0.0 & 0.0 \\ & & 4.832\times10^{9} & 0.0 & 0.0 & 0.0 \\ & & & 3.414\times10^{8} & 0.0 & 0.0 \\ & & \textit{symmetry} & & 4.585\times10^{8} & 0.0 \\ & & & & & 4.585\times10^{8} \end{bmatrix} \tag{2.1}$$

Here, there are some off-diagonal terms which are very small (in the order of $10^1 - 10^2$) comparing to the diagonal terms. Hence, they are considered as numerical noises and are not reported in the above matrix. The output file by VABS-IDE is provided in Appendix A for the readers to get a feeling of VABS-IDE outputs.

The corresponding 6×6 Timoshenko stiffness matrix generated by VABS-GUI is listed as follows:

$$\begin{bmatrix} 1.834 \times 10^{10} & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ & 4.682 \times 10^{9} & 0.0 & 0.0 & 0.0 & 0.0 \\ & & 4.682 \times 10^{9} & 0.0 & 0.0 & 0.0 \\ & & & 3.515 \times 10^{8} & 0.0 & 0.0 \\ & & \mathit{symmetry} & & 4.586 \times 10^{8} & 0.0 \\ & & & & & 4.586 \times 10^{8} \end{bmatrix} \tag{2.2}$$

Here again, off-diagonal terms which are very small (in the order of $10^1 - 10^2$) comparing to the diagonal terms are considered as numerical noises and are not reported. The output file by VABS-GUI is provided in Appendix B for the readers to get a feeling of VABS-GUI outputs.

Here, one thing worthy of notice is that there is a relatively large off-diagonal term S_{23} in the stiffness matrix computed by VABS-IDE. This unexpected term can be explained by the fact that the quality of the mesh generated by VABS-IDE is not as good as VABS-GUI. The circular cross-section is perfectly axisymmetric, yet the finite element mesh created by VABS-IDE is less symmetric, which is also disclosed by the fact that the transverse shear stiffnesses (S_{22} and S_{33}) along two directions, which should be the same, are also slightly different. There are two ways to alleviate the anomaly. One way is to try to force the mesh generator to create a finite element mesh featuring the same symmetry as the cross-section. Observing the meshes in Fig. 2.3 and Fig. 2.4, one can tell that the mesh created by VABS-GUI is more symmetric than that created by VABS-IDE. Usually, there is no much control a user can applied to a mesh generator, particularly when it is carrying out free meshing. The another way is to refine the mesh so that the anomaly will be alleviated by the increasing number of elements. For this reason, we refine the mesh and try to see whether the off-diagonal terms which are supposed to be zero to become smaller and smaller.

Table 2.2: Mesh data for generating the second mesh of aluminum tube using VABS-IDE

Heal toler	Max size	Min size	Fineness	Element type	# of Elements
0.01	0.025	0.018	User Defined	TRI3	553

We used the mesh control parameters in Table 2.2 to refine the mesh in VABS-IDE to generate a mesh of 553 elements as shown in Fig. 2.5.

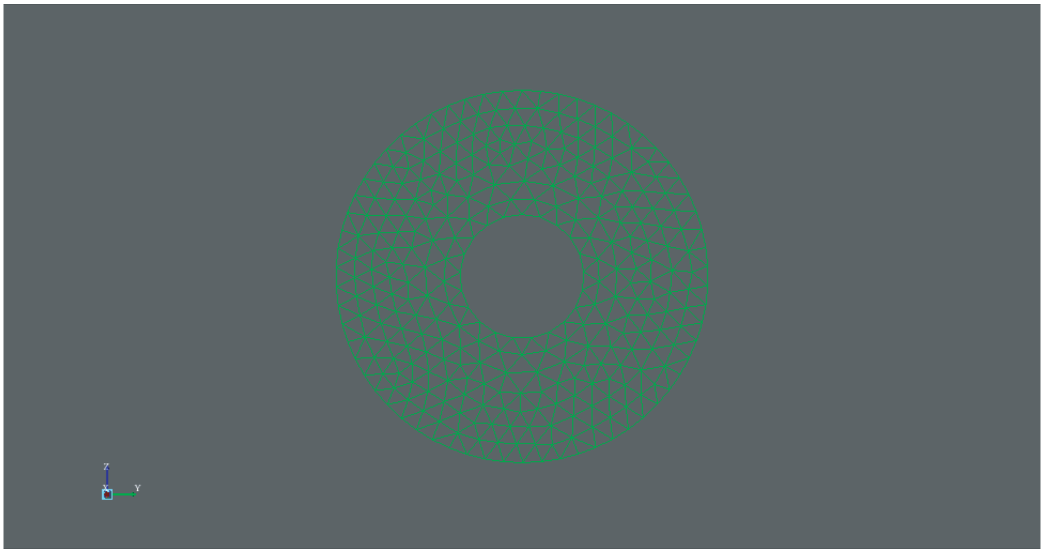


Fig. 2.5: second Mesh for Circular Aluminium Tube (553 elements, VABS-IDE)

To refine the mesh of VABS-GUI, one needs to click "Refine by splitting" three times to obtain a mesh of 577 elements as shown in Fig. 2.6.

The corresponding 6×6 Timoshenko stiffness matrix generated using VABS-IDE based on the mesh in Fig. 2.5 is:

$$\begin{bmatrix} 1.835 \times 10^{10} & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ & 4.726 \times 10^{9} & -1.851 \times 10^{5} & 0.0 & 0.0 & 0.0 \\ & & 4.727 \times 10^{9} & 0.0 & 0.0 & 0.0 \\ & & & 3.439 \times 10^{8} & 0.0 & 0.0 \\ & & \mathit{symmetry} & & 4.581 \times 10^{8} & 0.0 \\ & & & & & 4.581 \times 10^{8} \end{bmatrix} \tag{2.3}$$

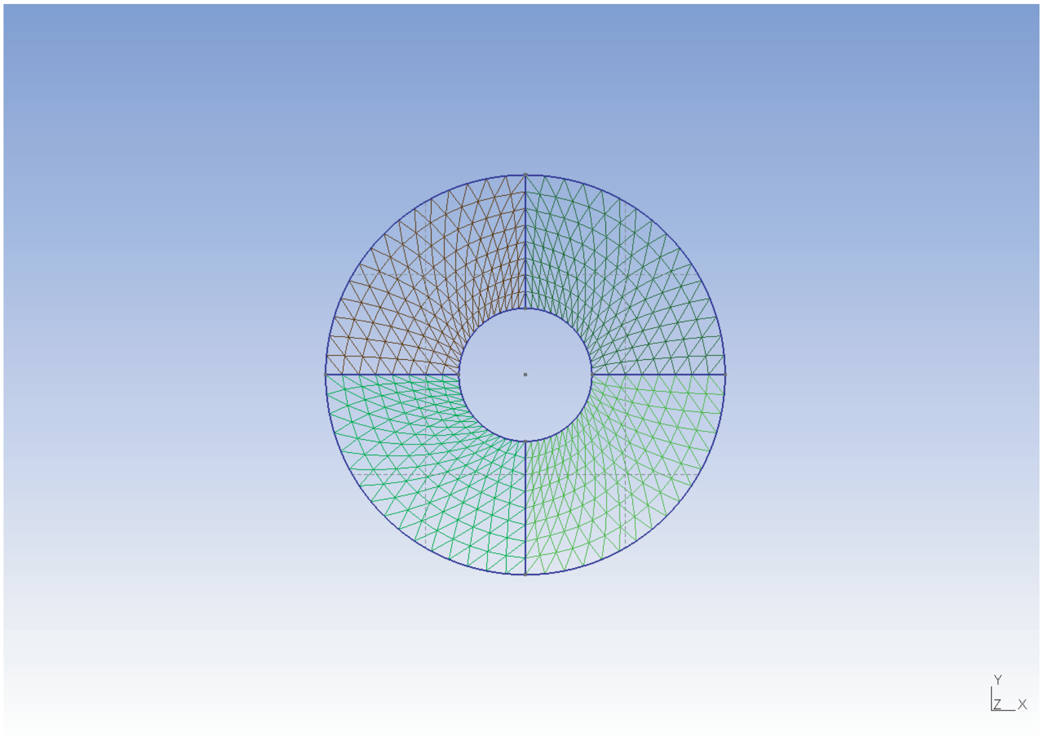


Fig. 2.6: Second Mesh for Circular Aluminium Tube (577 elements, VABS-GUI)

Comparing to those in Eq. (2.1), we notice that the diagonal terms remains almost the same. However, the unexpected off-diagonal term S_{23} changes significantly even with a different sign. This clearly confirms that this anomaly is associated with the mesh.

The corresponding 6×6 Timoshenko stiffness matrix generated using VABS-GUI based on the mesh in Fig. 2.6 is:

$$\begin{bmatrix} 1.834\times10^{10} & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ & 4.695\times10^{9} & 0.0 & 0.0 & 0.0 & 0.0 \\ & & 4.695\times10^{8} & 0.0 & 0.0 & 0.0 \\ & & & 3.515\times10^{8} & 0.0 & 0.0 \\ & & \mathit{symmetry} & & 4.586\times10^{8} & 0.0 \\ & & & & & 4.586\times10^{8} \end{bmatrix} \tag{2.4}$$

Table 2.3: Mesh data for generating the second mesh of aluminum tube using VABS-IDE

Heal toler	Max size	Min size	Fineness	Element type	# of Elements
0.01	0.015	0.008	User Defined	TRI3	2179

To refine the mesh further, we use the mesh control parameters as listed in Table 2.3 to generate a mesh of 2179 elements in total as shown in Fig. 2.7

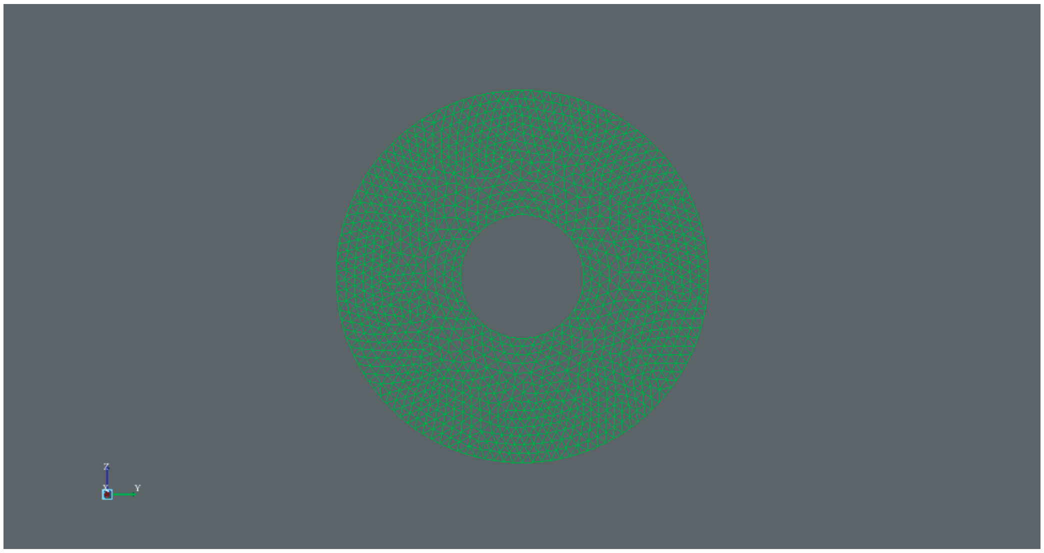


Fig. 2.7: Third Mesh of Circular Aluminium Tube (2179 elements, VABS-IDE)

To refine the mesh using VABS-GUI, one needs to click “Refine by splitting” four times. A finite element of the cross containing a total 2177 element is generated as shown in Fig. 2.8.

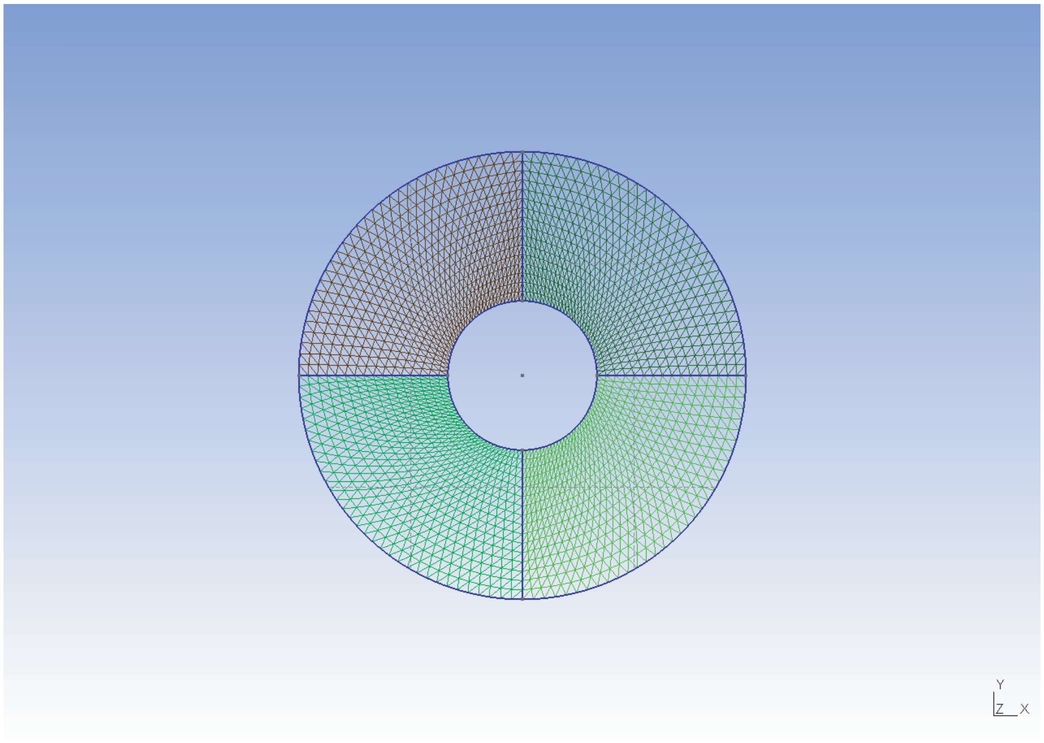


Fig. 2.8: Third Mesh of Circular Aluminium Tube (2177 elements, VABS-GUI)

VABS-IDE produces the following 6 × 6 Timoshenko stiffness matrix according to the mesh in Fig. 2.7:

$$\begin{bmatrix} 1.834\times10^{10} & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ & 4.716\times10^{9} & 7.288\times10^{3} & 0.0 & 0.0 & 0.0 \\ & & 4.727\times10^{9} & 0.0 & 0.0 & 0.0 \\ & & & 3.496\times10^{8} & 0.0 & 0.0 \\ & & \mathit{symmetry} & & 4.586\times10^{8} & 0.0 \\ & & & & & 4.586\times10^{8} \end{bmatrix} \tag{2.5}$$

Clearly the off-diagonal is getting significantly smaller in comparison to the previous two results and the diagonal terms. Indeed with the increasing of number of elements, the off-diagonal term which is supposed to be zero is getting smaller.

VABS-GUI produces the following 6 × 6 Timoshenko stiffness matrix according to the mesh in Fig. 2.8:

$$\begin{bmatrix} 1.834\times10^{10} & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ & 4.683\times10^{9} & 0.0 & 0.0 & 0.0 & 0.0 \\ & & 4.683\times10^{9} & 0.0 & 0.0 & 0.0 \\ & & & 3.515\times10^{8} & 0.0 & 0.0 \\ & & \textit{symmetry} & & 4.586\times10^{8} & 0.0 \\ & & & & & 4.586\times10^{8} \end{bmatrix} \tag{2.6}$$

Although it is generally true with VABS-IDE that with increasing number of elements, the off-diagonal terms which are supposed to be zero are getting smaller. However, it will be much more accurate and efficient if the mesh generator can take advantage of the symmetry of the real geometry to create a high-quality like what VABS-GUI does. We have also tested that even if we increases to total number of elements of the VABS-IDE mesh as large as 14131, the off-diagonal term still is around the order of 10^3. This shows that the qualify of the mesh generated by VABS-IDE is not as good as VABS-GUI.

For reference, the Timoshenko stiffness matrix of the same cross-section with inputs prepared by PreVABS published in [1] is listed below:

$$\begin{bmatrix} 1.834\times10^{10} & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ & 4.682\times10^{9} & 0.0 & 0.0 & 0.0 & 0.0 \\ & & 4.682\times10^{9} & 0.0 & 0.0 & 0.0 \\ & & & 3.515\times10^{8} & 0.0 & 0.0 \\ & & \textit{symmetry} & & 4.586\times10^{8} & 0.0 \\ & & & & & 4.586\times10^{8} \end{bmatrix} \tag{2.7}$$

which has been verified to reproduce the exact solution for EA, GJ, EI_{22}, EI_{33} according to the linear elasticity theory.

Table 2.4 lists the results obtained by using VABS-IDE (Eq. (2.5)) and VABS-GUI

(Eq. (2.6)). Comparing the stiffness results predicted by VABS using inputs provided by VABS-IDE, VABS-GUI, and PreVABS. The results from VABS-GUI and PreVABS are almost identical. However, there is a slight loss of accuracy for VABS-IDE results, particularly the torsional stiffness and transverse shear stiffness. This can be explained by the fact that these values must be obtained through solving a set of partial differential equations which poses more demanding requirement for the quality of mesh. Nevertheless, all the results are agreeing with each other very well without any significant differences. Since this is a very simple isotropic cross-section with simple geometry, when use all these three preprocessing tools to generate the mesh, none of them will hardly loss the accuracy. That is why we can obtain highly accurate results.

Table 2.4: Comparison of stiffness results for the first example

	VABS-IDE	VABS-GUI	PreVABS	% Diff.(IDE)	% Diff.(GUI)
EA	1.834E+10	1.834E+10	1.834E+10	0	0
GJ	3.496E+08	3.515E+08	3.515E+08	-0.7%	0
EI_{22}	4.586E+08	4.586E+08	4.586E+08	0	0
EI_{33}	4.586E+08	4.586E+08	4.586E+08	0	0
S_{22}	4.716E+09	4.683E+09	4.682E+09	0.7%	0
S_{33}	4.727E+09	4.683E+09	4.682E+09	1.0%	0

2.2 A channel section

The second example is a channel section (see Fig. 2.9) made of an isotropic material with E=206.843 GPa, $\nu = 0.49$ and $\rho = 1068.69$ kg/m^3. It is a typical thin-walled section with its beam stiffness can be analytically obtained using the thin-walled theory. In Ref. [1], this channel section was modeled as a highly heterogeneous section as shown in Fig. 2.10 artificially made from the isotropic channel section plus a fake material with its Young's modulus and density $\rho = 10^{-12}$ times smaller than those of the real material. It is expected that the fake material will not provide stiffness and inertia to this section because of its extremely small modulus and density. Hence, the overall properties will be the same as the isotropic channel section, whose analytical solution can be readily obtained using the thin-walled theory. This was done because PreVABS was limited to preprocess sections with airfoil type external geometry. The adding of fake material not only adds approximation to the results but also increases the finite element model size of the overall cross-section. With VABS-IDE and VABS-GUI, we can model this section directly without introducing the fake materials.

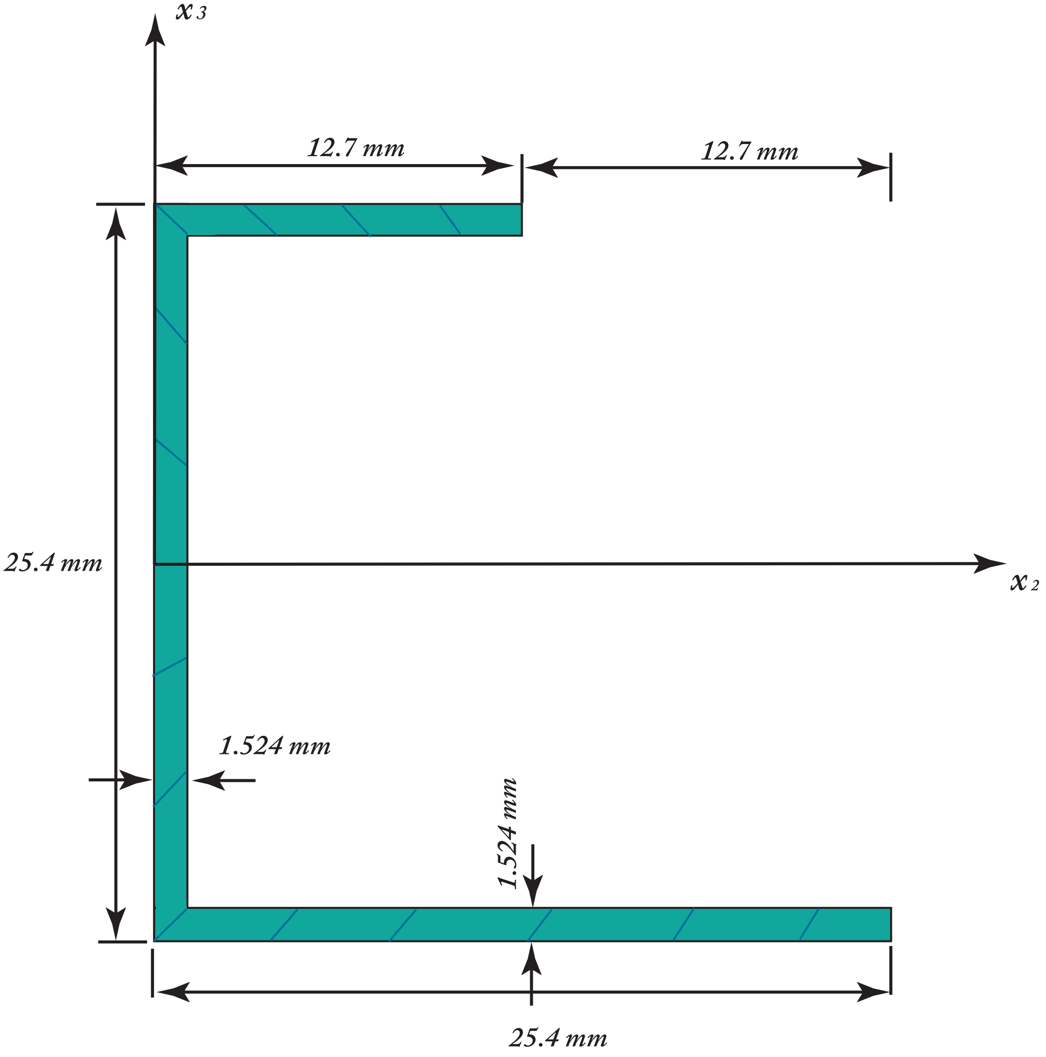


Fig. 2.9: A Sketch of the Isotropic Channel Section

Using the mesh control parameters in Table 2.5 in VABS-IDE, we can generate a finite element mesh of a total of 2420 three-noded triangular elements as shown in Fig. 2.11. Note here we have select our mesh control parameters in such a way that those terms which are supposed to be zero are much smaller than those terms which are not zero in the stiffness

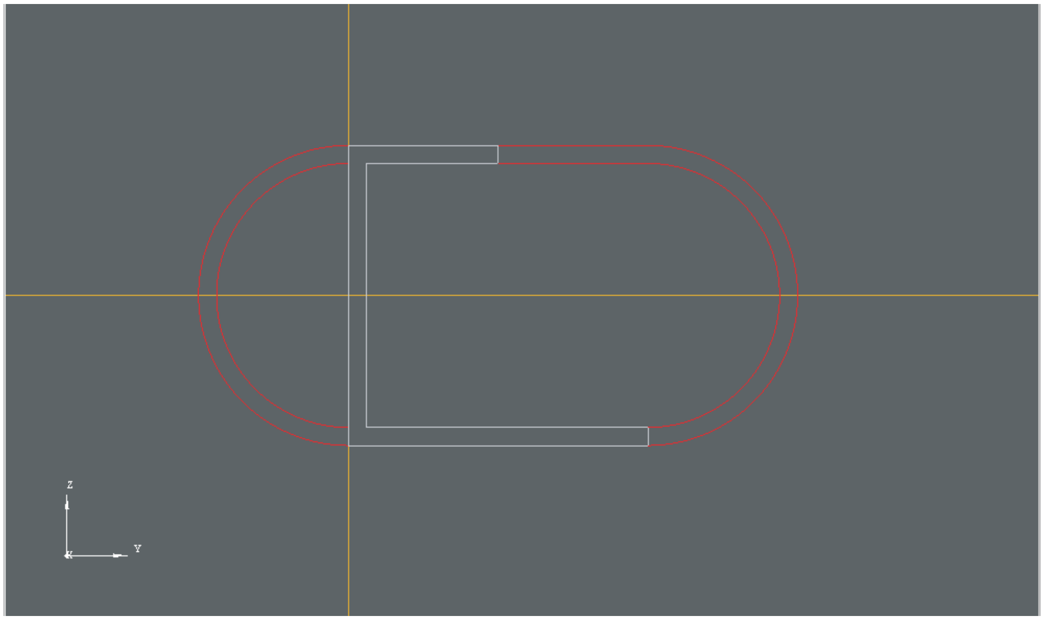


Fig. 2.10: The Highly Heterogeneous Section Used to Model the Channel Section in Ref. [1]

Table 2.5: Mesh data for generating the mesh of the channel section using VABS-IDE

Heal toler	Max size	Min size	Fineness	Element type	# of Elements
0.001	0.0003	0.0003	User Defined	TRI3	2420

matrix.

The 6 × 6 Timoshenko stiffness matrix generated by VABS-IDE is listed as follows:

$$\begin{bmatrix} 1.905\times10^{7} & 0.0 & 0.0 & 0.0 & -4.779\times10^{4} & -1.324\times10^{5} \\ & 2.788\times10^{6} & 2.373\times10^{5} & 2.119\times10^{4} & 0.0 & 0.0 \\ & & 2.137\times10^{6} & -7.697\times10^{3} & 0.0 & 0.0 \\ & & & 2.085\times10^{2} & 0.0 & 0.0 \\ & & \textit{symmetry} & & 2.010\times10^{3} & 9.106\times10^{2} \\ & & & & & 1.945\times10^{3} \end{bmatrix} \tag{2.8}$$

The same cross-section can be created and meshed in VABS-GUI as shown in Fig. 2.12 and Fig. 2.13.

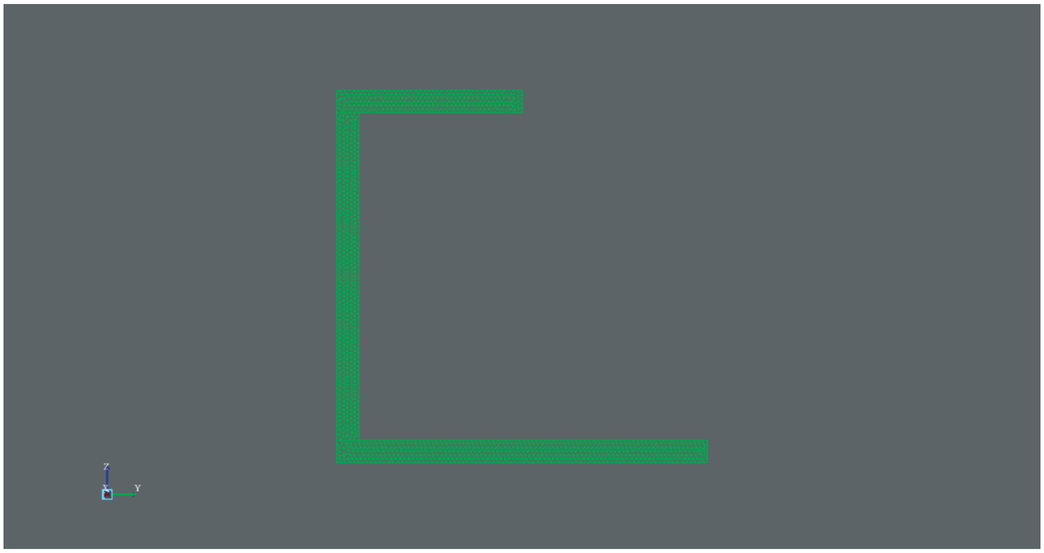


Fig. 2.11: Mesh for Channel Section (1101 elements, VABS-IDE)

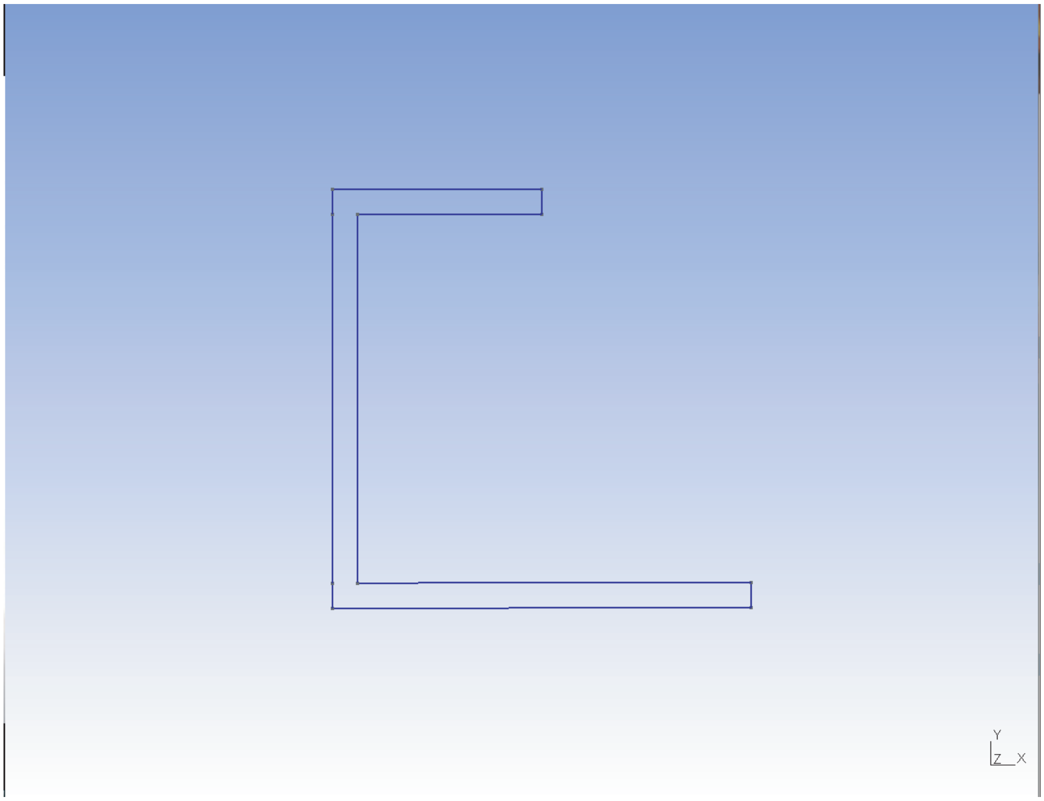


Fig. 2.12: Geometry of the Channel Section (VABS-GUI)

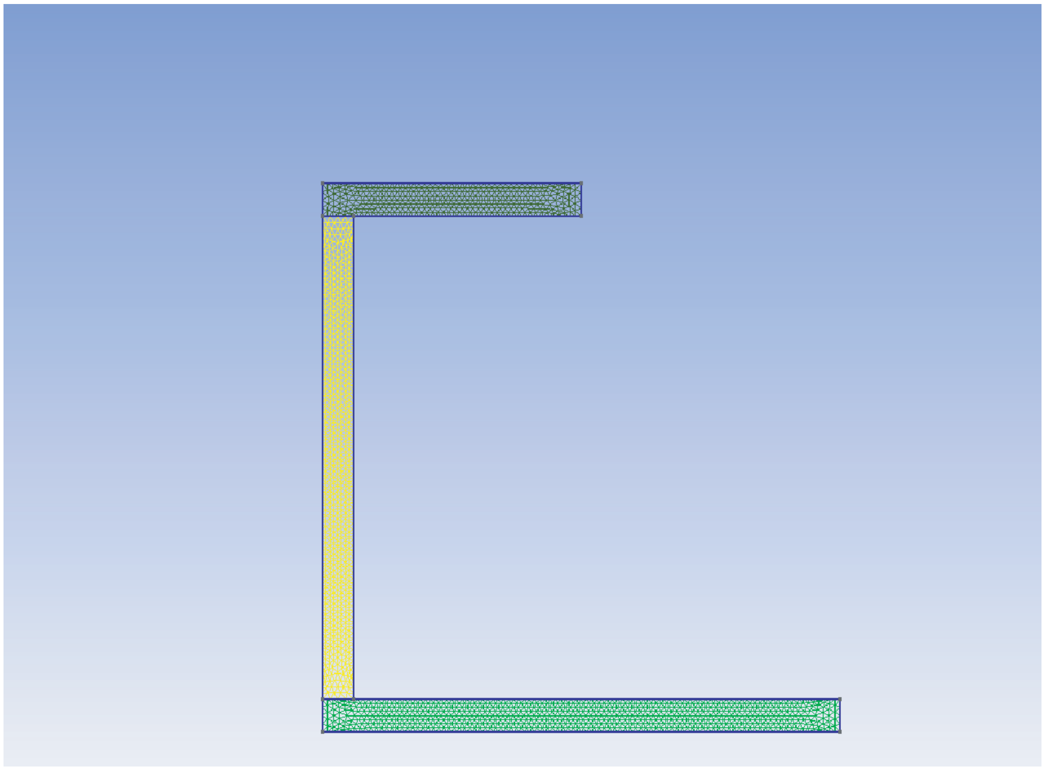


Fig. 2.13: Mesh of the Channel Section (2042 elements, VABS-GUI)

The corresponding 6×6 Timoshenko stiffness matrix generated by VABS-GUI is listed as follows:

$$\begin{bmatrix} 1.903 \times 10^{7} & 0.0 & 0.0 & 0.0 & -4.778 \times 10^{4} & -1.325 \times 10^{5} \\ & 2.791 \times 10^{6} & 2.365 \times 10^{5} & 2.120 \times 10^{4} & 0.0 & 0.0 \\ & & 2.137 \times 10^{6} & -7.679 \times 10^{3} & 0.0 & 0.0 \\ & & & 2.086 \times 10^{2} & 0.0 & 0.0 \\ & & \mathit{symmetry} & & 2.010 \times 10^{3} & 9.102 \times 10^{2} \\ & & & & & 1.945 \times 10^{2} \end{bmatrix} \tag{2.9}$$

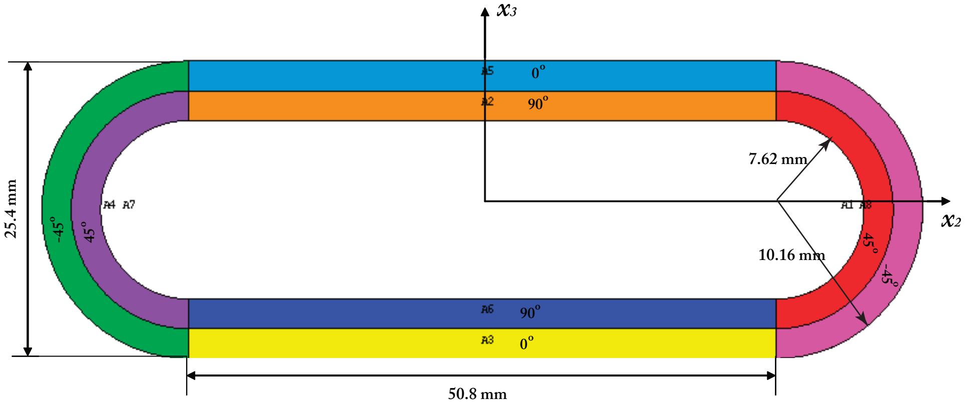


Fig. 2.14: Schematic of a Multilayer Composite Pipe

For reference, the Timoshenko stiffness matrix calculated by PreVABS in Ref. [1] is also listed below:

$$\begin{bmatrix} 1.903\times10^{7} & 0.0 & 0.0 & 0.0 & -4.778\times10^{4} & -1.325\times10^{5} \\ & 2.791\times10^{6} & 2.364\times10^{5} & 2.122\times10^{4} & 0.0 & 0.0 \\ & & 2.137\times10^{6} & -7.679\times10^{3} & 0.0 & 0.0 \\ & & & 2.086\times10^{2} & 0.0 & 0.0 \\ & & \textit{symmetry} & & 2.010\times10^{3} & 9.102\times10^{2} \\ & & & & & 1.944\times10^{3} \end{bmatrix} \tag{2.10}$$

It can be observed that all the three stiffness matrix generated for the same cross-section with the inputs prepared by VABS-IDE, VABS-GUI, and PreVABS agree with each other very well.

2.3 A multilayer composite pipe

The third example is a multi-layer composite pipe with the geometry and the lamination information shown in Fig. 2.14. It is a thin-walled cross-section with the thickness of wall to the chord length ration is less than 0.1. Each layer is made of composite materials having properties as E_{11}=141.963 GPa, E_{22}=E_{33}=9.79056 GPa, G_{12}=G_{13}=G_{23}=59.9844 GPa and ν_{12}=ν_{13}=ν_{23}=0.42.

Table 2.6: Mesh data for generating the mesh of the multilayer pipe section using VABS-IDE

Heal toler	Max size	Min size	Fineness	Element type	# of Elements
0.01	0.0006	0.0004	User Defined	TRI3	4800

The geometry created in VABS-IDE is shown in Fig. 2.15. Using the mesh control parameters listed in Table 2.6, VABS-IDE will create a finite element mesh for this cross-section with 4800 triangular elements as shown in Fig. 2.16.

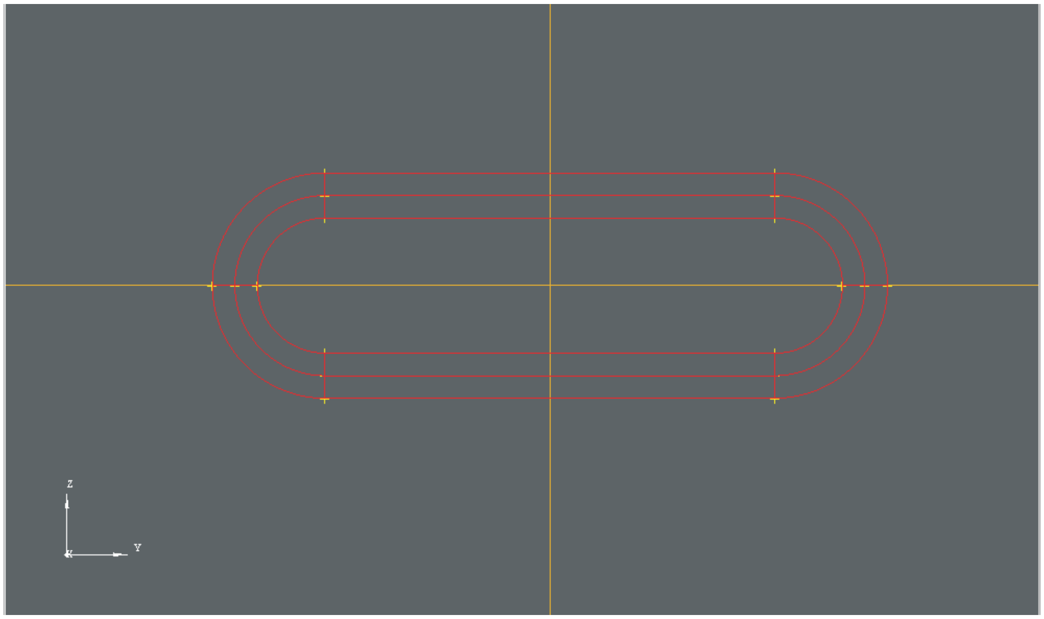


Fig. 2.15: Geometry of Multilayer Composite Pipe (VABS-IDE)

The 6 × 6 Timoshenko stiffness matrix generated by VABS-IDE is listed as follows:

$$\begin{bmatrix}
4.623\times10^{7} & 0.0 & 0.0 & 1.113\times10^{4} & 0.0 & 0.0\\
 & 3.467\times10^{6} & 0.0 & 0.0 & -9.254\times10^{2} & 0.0\\
 & & 1.376\times10^{6} & 0.0 & 0.0 & -5.861\times10^{3}\\
 & & & 1.970\times10^{3} & 0.0 & 0.0\\
 & & \textit{symmetry} & & 5.415\times10^{3} & 0.0\\
 & & & & & 1.540\times10^{4}
\end{bmatrix} \tag{2.11}$$

Fig. 2.17 and Fig. 2.18 are geometry and mesh, respectively, generated by VABS-GUI.

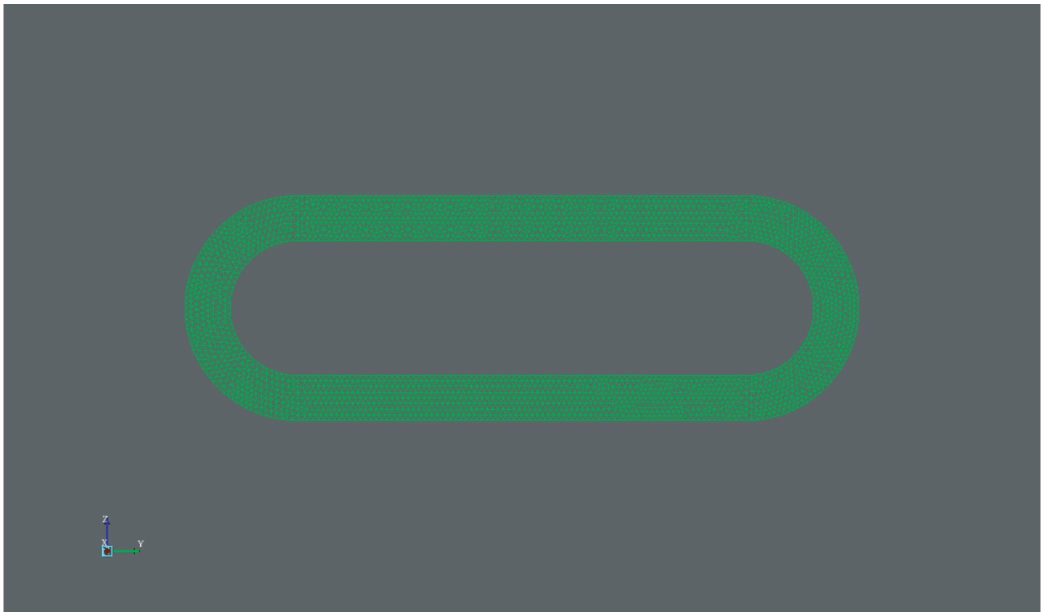


Fig. 2.16: Mesh for Multilayer Composite Pipe (4800 elements, VABS-IDE)

The mesh has a total of 4602 elements which is obtained by clicking “Refine by splitting” twice.

Then, 6 × 6 Timoshenko stiffness matrix generated by VABS-GUI is listed as below:

$$\begin{bmatrix} 4.623\times10^{7} & 0.0 & 0.0 & 1.113\times10^{4} & 0.0 & 0.0 \\ & 3.467\times10^{6} & 0.0 & 0.0 & -9.254\times10^{2} & 0.0 \\ & & 1.376\times10^{6} & 0.0 & 0.0 & -5.858\times10^{3} \\ & & & 1.973\times10^{4} & 0.0 & 0.0 \\ & & \mathit{symmetry} & & 5.405\times10^{3} & 0.0 \\ & & & & & 1.547\times10^{4} \end{bmatrix} \tag{2.12}$$

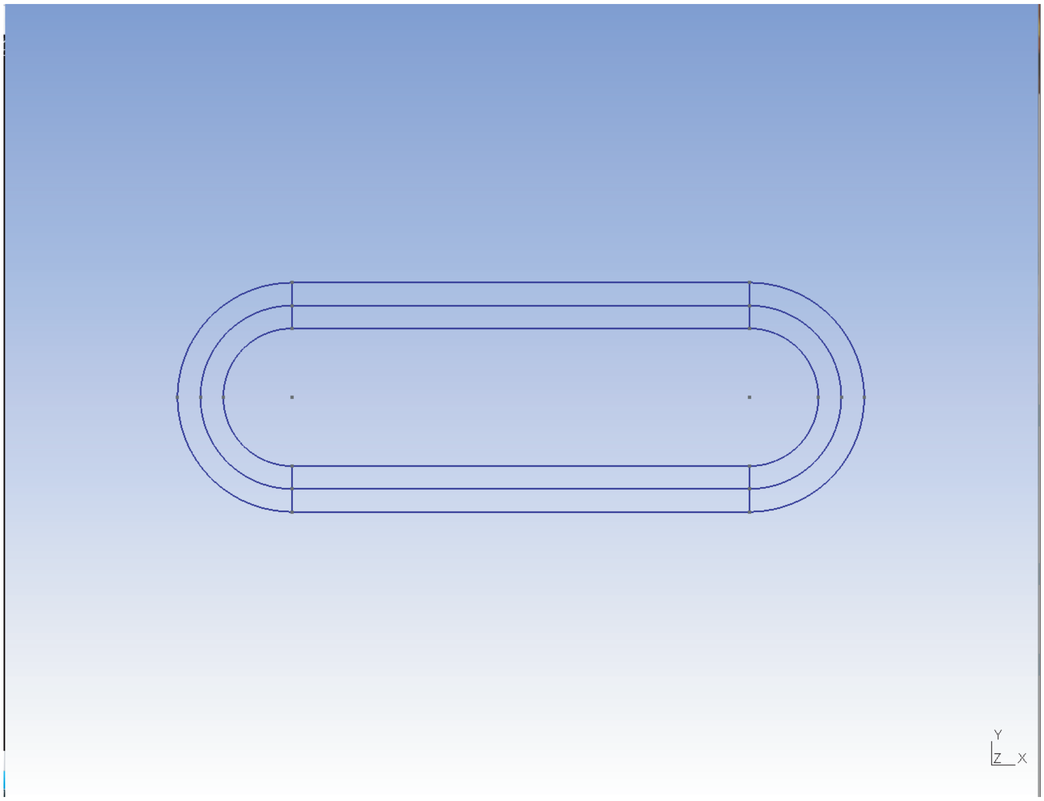


Fig. 2.17: Geometry of the Multilayer Composite Pipe (VABS-GUI)

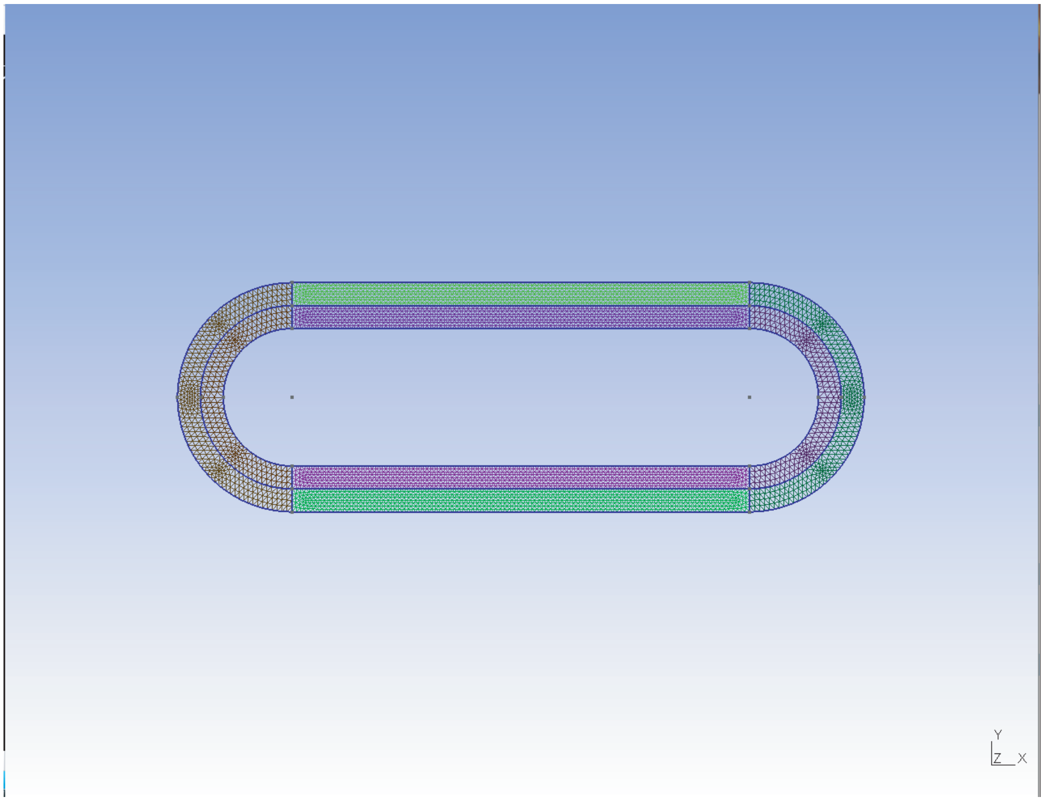


Fig. 2.18: Mesh for Multilayer Composite Pipe (4602 elements, VABS-GUI)

For reference, the Timoshenko stiffness matrix generated by PreVABS according to Ref. [1] is also listed here:

$$\begin{bmatrix} 4.621\times10^{7} & 0.0 & 0.0 & 1.111\times10^{4} & 0.0 & 0.0 \\ & 3.489\times10^{6} & 0.0 & 0.0 & -9.251\times10^{2} & 0.0 \\ & & 1.463\times10^{6} & 0.0 & 0.0 & -5.859\times10^{3} \\ & & & 1.971\times10^{4} & 0.0 & 0.0 \\ & & \textit{symmetry} & & 5.402\times10^{3} & 0.0 \\ & & & & & 1.547\times10^{4} \end{bmatrix} \tag{2.13}$$

Table 2.7: Comparison of stiffness results for the multilayer composite pipe

	VABS-IDE	VABS-GUI	PreVABS	% Diff.(IDE)	% Diff.(GUI)
EA	4.623E+07	4.623E+07	4.621E+07	0.04	0.04
GJ	1.970E+04	1.973E+04	1.971E+04	-0.05	0.1
EI_{22}	5.415E+03	5.405E+03	5.402E+03	0.2	0.06
EI_{33}	1.540E+04	1.547E+04	1.547E+04	-0.5	0
S_{22}	3.467E+06	3.467E+06	3.489E+06	-0.6	-0.6
S_{33}	1.376E+06	1.376E+06	1.463E+06	6	6
S_{14}	1.113E+04	1.113E+04	1.111E+04	0.18	0.18
S_{25}	-9.254E+02	-9.254E+02	-9.251E+02	0.03	0.03
S_{36}	-5.861E+03	-5.858E+03	-5.859E+03	0.03	0.02

This cross-section is more complex than the previous two examples, and the difference between VABS-IDE and VABS-GUI and PreVABS is getting slightly bigger. Particularly, for the transverse shear stiffness S_{33}, there are around 6% difference between VABS-IDE and VABS-GUI results in comparison to PreVABS results. More investigation is needed to find out why the difference is bigger for this term. Particularly, a close look at the geometry and the mesh produced using PreVABS published in Ref. [1] is necessary to find out why there is bigger difference for this particular value.

2.4 An isotropic blade-like section

The fourth example is an isotropic blade-like section as shown in Fig. 2.19. It is noted that the inclined straight edges at the tail are tangent to the ending arc at the head. Material

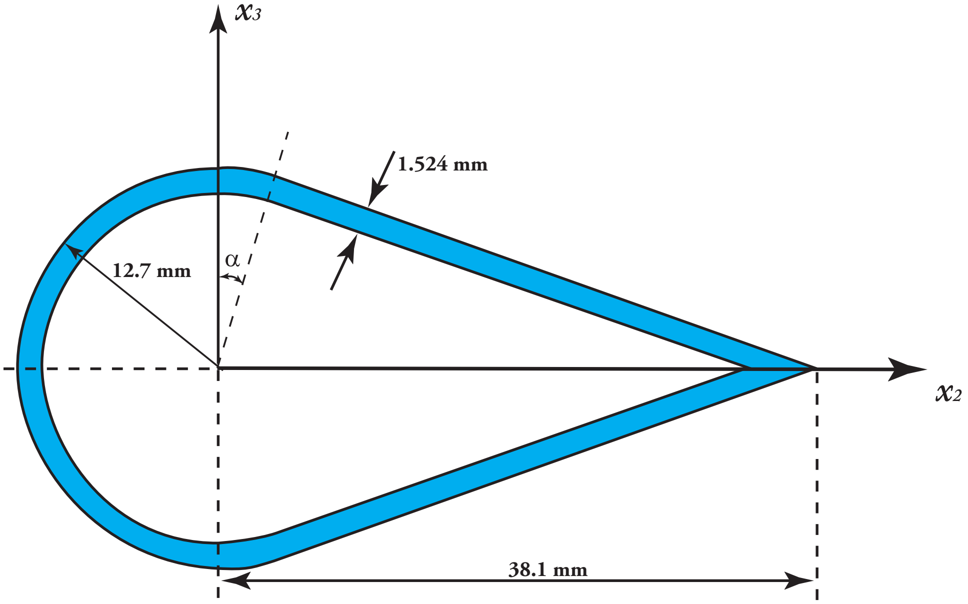


Fig. 2.19: Schematic of an isotropic blade-like section.

Table 2.8: Mesh data for generating mesh for isotropic blade-like section in VABS-IDE

Heal toler	Max size	Min size	Fineness	Element type	# of Elements
0.01	0.0012	0.0012	User Defined	TRI3	980

properties of this section are the same as the second example (E=206.843 GPa, $\nu = 0.49$, and $\rho = 1068.69\text{kg/m}^3$).

Using the mesh control parameters in Table 2.8, we can create a finite element mesh of 980 elements in VABS-IDE as shown in Fig. 2.20.

The corresponding Timoshenko stiffness matrix generated by VABS-IDE is as follows:

$$\begin{bmatrix} 3.701 \times 10^7 & 0.0 & 0.0 & 0.0 & 0.0 & -3.401 \times 10^5 \\ & 8.737 \times 10^6 & 0.0 & 0.0 & 0.0 & 0.0 \\ & & 2.623 \times 10^6 & 2.375 \times 10^3 & 0.0 & 0.0 \\ & & & 1.828 \times 10^3 & 0.0 & 0.0 \\ & & \textit{symmetry} & & 2.222 \times 10^3 & 0.0 \\ & & & & & 1.139 \times 10^4 \end{bmatrix} \tag{2.14}$$

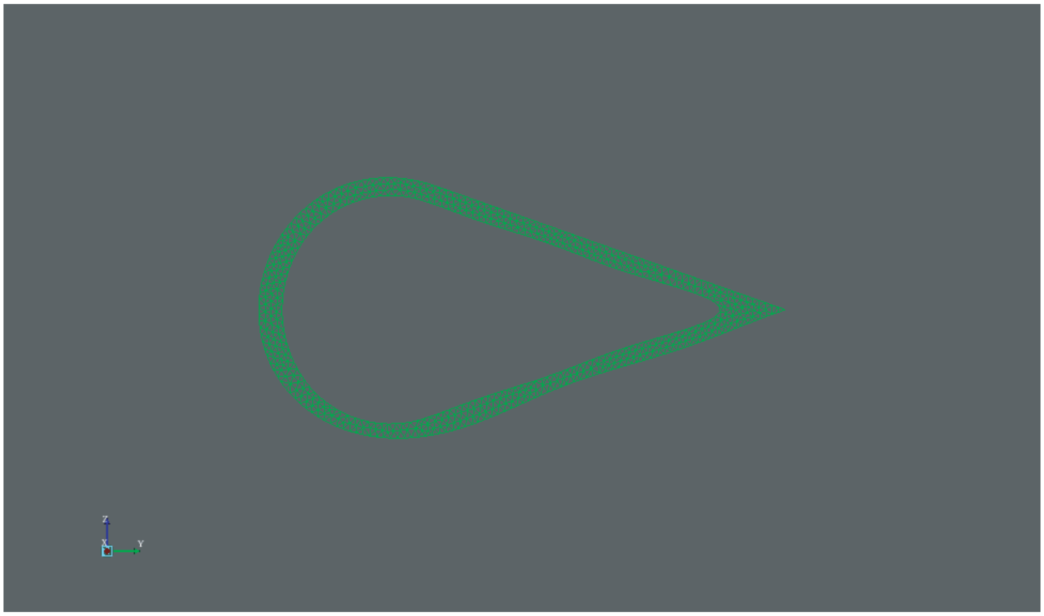


Fig. 2.20: Mesh for Isotropic Blade-like Section (960 elements, VAB-IDE)

Similarly using VABS-GUI, we can create the same cross-section and mesh it. The corresponding finite element mesh is shown in Fig. 2.21 which has a total of 1045 elements by clicking “Refine by splitting” twice.

The corresponding 6 × 6 Timoshenko stiffness matrix generated by VABS-GUI is

$$\begin{bmatrix} 3.566\times10^{7} & 0.0 & 0.0 & 0.0 & 0.0 & -3.391\times10^{5} \\ & 8.253\times10^{6} & 0.0 & 0.0 & 0.0 & 0.0 \\ & & 2.444\times10^{6} & 2.382\times10^{3} & 0.0 & 0.0 \\ & & & 1.762\times10^{3} & 0.0 & 0.0 \\ & & \textit{symmetry} & & 2.101\times10^{3} & 0.0 \\ & & & & & 1.113\times10^{4} \end{bmatrix} \tag{2.15}$$

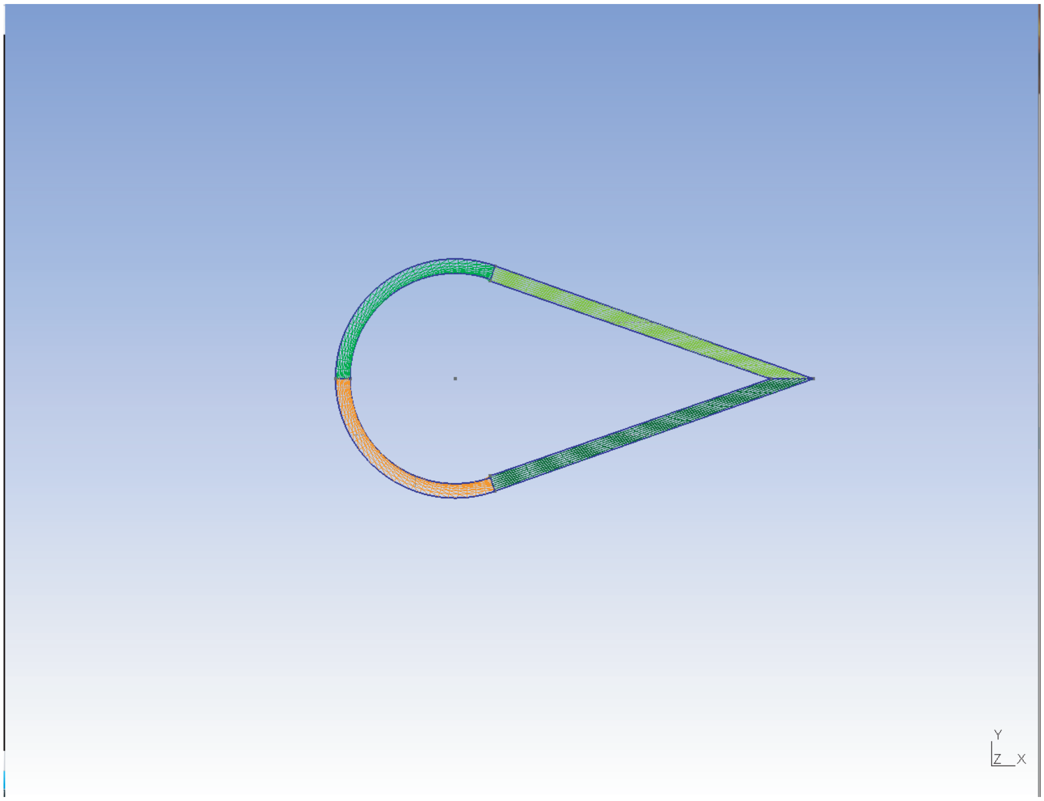


Fig. 2.21: Mesh for Isotropic Blade-like Section (1045 elements, VABS-GUI)

For reference, the Timoshenko stiffness matrix obtained by PreVABS according to Ref. [1] is listed below:

$$\begin{bmatrix} 3.566\times 10^{7} & 0.0 & 0.0 & 0.0 & 0.0 & -3.394\times 10^{5} \\ & 8.252\times 10^{6} & 0.0 & 0.0 & 0.0 & 0.0 \\ & & 2.444\times 10^{6} & 2.382\times 10^{3} & 0.0 & 0.0 \\ & & & 1.762\times 10^{3} & 0.0 & 0.0 \\ & & \textit{symmetry} & & 2.101\times 10^{3} & 0.0 \\ & & & & & 1.113\times 10^{4} \end{bmatrix} \tag{2.16}$$

These stiffness results generated by VABS-IDE, VABS-GUI, and PreVABS are also compared in Table 2.9, where the percentage difference is also computed for each nonzero stiffness terms. In this example, I need to emphasize here that there is a bug of VABS-IDE. I can generate the exact cross-sectional geometry by correctly creating all the points, connecting the straight lines and arcs. However, when I mesh the cross-section in VABS-IDE, the geometry of the plane is changed slightly due to mesh which is a very strange behavior of VABS-IDE currently debugging by Advanced Dynamics Inc., the developer of VABS-IDE. This partially explains why the results generated by VABS-IDE are different from those generated by PreVABS. On the other hand, VABS-GUI has not such error and the results are almost the same those those generated by PreVABS, which was also validated using thin-walled theory.

Table 2.9: Comparison of results for isotropic blade-like section

	VABS-IDE	VABS-GUI	PreVABS	% Diff.(IDE)	% Diff.(GUI)
EA	3.701E+07	3.566E+07	3.566E+07	3.78	0
GJ	1.828E+03	1.762E+03	1.762E+03	3.74	0
EI_{22}	2.222E+03	2.101E+03	2.101E+03	5.75	0
EI_{33}	1.139E+04	1.113E+04	1.113E+04	2.33	0
S_{22}	8.737E+06	8.253E+06	8.252E+06	5.8	0.01
S_{33}	2.623E+06	2.444E+06	2.444E+06	7.3	0
S_{34}	2.375E+03	2.382E+03	2.382E+03	0.29	0
S_{16}	-3.401E+05	-3.391E+05	-3.394E+05	0.2	0.09

Chapter 3

Smeared Properties and Mesh Convergence

In this chapter, we will first use a simple composite strip example to demonstrate the loss of accuracy using smeared properties in modeling composite beams. Then we will carry out a mesh convergence study of the sensitivity of VABS results with respect to the number of elements used to mesh the cross-section. As the example use in this study is simple, and we have verified that VABS-IDE, VABS-GUI, and PreVABS generate similar results for simple cross-sections. Hence, in this chapter all the cases are analyzed using VABS-IDE due to convenience.

3.1 Smeared properties

In modeling composite beams in both helicopter and wind turbine industry, the so-called smeared properties is frequently employed due to two reasons:

- In the past, the industry does not have access to a sophisticated analysis tool as VABS to carry out a layer-by-layer detailed modeling of composite beams. Usually, some proprietary tools based on simplified analytical formulas are used to model composite beams. These tools can only take in smeared properties.

- For modern composite slender structures, such as composite helicopter rotor blades, there are possibly hundreds of layers in the cross-section, and smeared properties are usually used to simplify the geometry and finite element modeling.

However, what will be the loss of accuracy due to using smeared properties was not quantified due to the lack of a high-fidelity analysis tool as VABS to model all the details of material layers. This report will present the first attempt to access the loss of accuracy due to smeared properties using a simple example.

The motivation of using smeared properties is to reduce the number of layers in the composite beam needed for modeling so that the corresponding finite element could have less number of elements. For example for a 8-layer rectangular composite beam, the lay-up sequence is

$$\left[\ 25_2 \ \middle|\ 50 \ \middle|\ 0 \ \middle|\ 50 \ \middle|\ 0 \ \middle|\ 25_2\ \right]_T$$

All the layers have equal thickness of 0.25 mm, with total thickness as 2 mm and the width as 8 cm. The layers are made of a single composite material with E_{11}=41.5 GPa, E_{22}=E_{33}=7.83 GPa, $\nu_{12} = \nu_{13} = \nu_{23} = 0.3$, G_{12}=3.15 GPa, $G_{13} = G_{23} = 3.01$ GPa

This simple example of course can be simply modeled with all the details of each layer using VABS-IDE with the corresponding geometry and mesh shown in Fig. 3.1 and Fig. 3.2, respectively.

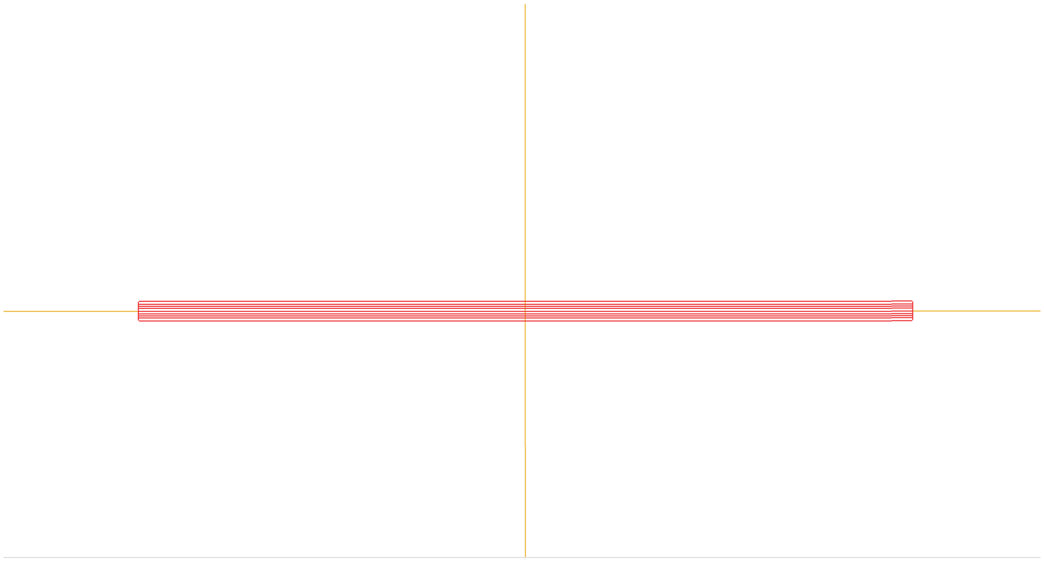


Fig. 3.1: Geometry of a Simple Composite Strip

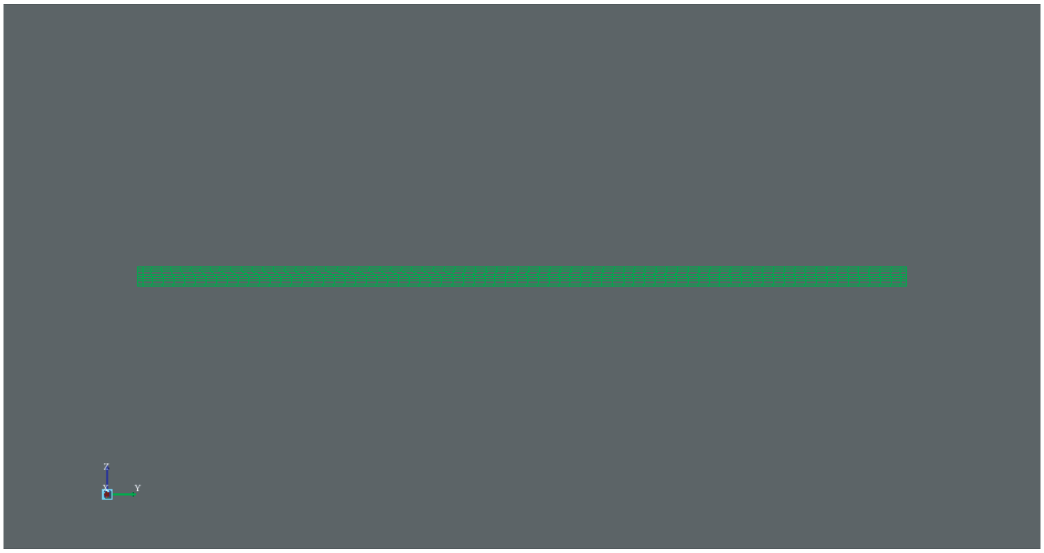


Fig. 3.2: Mesh for A Simple Composite Strip

The Timoshenko stiffness matrix generated by VABS-IDE based on this mesh is as follows:

$$\begin{bmatrix} 4.108\times10^{6} & 2.438\times10^{3} & -1.370\times10^{1} & -1.096\times10^{3} & -1.630\times10^{2} & -6.747\times10^{1} \\ & 9.909\times10^{5} & 1.379\times10^{2} & -1.031\times10^{2} & 6.002\times10^{2} & 3.912\times10^{1} \\ & & 1.488\times10^{4} & 1.467\times10^{-1} & -2.65\times10^{-1} & 3.025\times10^{0} \\ & & & 1.523\times10^{0} & -3.868\times10^{-2} & 3.672\times10^{-2} \\ & \text{symmetry} & & & 1.509E+\times10^{0} & -2.553\times10^{-2} \\ & & & & & 2.173\times10^{3} \end{bmatrix} \tag{3.1}$$

The same problem can be analyzed using smeared properties if we try to avoid modeling the cross-section layer by layer. We basically assume that the cross-section is made of a single layer with the material properties is obtained as the average of the material properties of all the layers. To use smeared properties, we need to perform the following steps:

- Obtain the 3-D material properties of each layer expressed in a single coordinate system. Usually we choose the beam coordinate system.

- Calculate the average of the 3-D material properties weighted using the layer thickness.

- Use the averaged 3-D material properties to carry out the cross-sectional analysis.

Here, the critical step is to perform the transformation. Let us present the general formula for how to do this transformation. We know for linear elastic material, the constitutive relation obeys the generalized Hooke's law as follows:

$$\sigma = C\varepsilon$$

where $\sigma = \lfloor \sigma_{11} \quad \sigma_{22} \quad \sigma_{33} \quad \sigma_{23} \quad \sigma_{13} \quad \sigma_{12} \rfloor$, a column matrix containing the stress tensor, $\varepsilon = \lfloor \varepsilon_{11} \;\; \varepsilon_{22} \;\; \varepsilon_{33} \;\; 2\varepsilon_{23} \;\; 2\varepsilon_{13} \;\; 2\varepsilon_{12} \rfloor$, a column matrix containing the strain tensor, and C is the 6×6 stiffness matrix and for othotropic materials, C can be expressed as

$$C = \begin{bmatrix} C_{11} & C_{12} & C_{13} & 0 & 0 & 0 \\ C_{21} & C_{22} & C_{23} & 0 & 0 & 0 \\ C_{31} & C_{32} & C_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{66} \end{bmatrix} \tag{3.2}$$

with the corresponding compliance matrix as:

$$S = C^{-1}$$

with

$$S = \begin{bmatrix} 1/E_1 & -\nu_{12}/E_1 & -\nu_{13}/E_1 & 0 & 0 & 0 \\ -\nu_{21}/E_2 & 1/E_2 & -\nu_{23}/E_1 & 0 & 0 & 0 \\ -\nu_{31}/E_2 & -\nu_{32}/E_3 & 1/E_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/G_{23} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/G_{31} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/G_{12} \end{bmatrix} \tag{3.3}$$

$$\nu_{21} = \nu_{12} * E_2/E_1$$

$$\nu_{31} = \nu_{13} * E_3/E_1$$

$$\nu_{32} = \nu_{23} * E_3/E_2$$

These expressions are in the material coordinate system. If the material coordinate system can be brought to the beam coordinate system by first rotating around the third axis θ_3, then rotating around the second axis θ_2, and then around the first axis θ_1, the direction cosine matrix relating the beam coordinate system and the material coordinate system can be obtained as

$$\beta = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_1 & \sin\theta_1 \\ 0 & \sin\theta_1 & \cos\theta_1 \end{pmatrix} \begin{pmatrix} \cos\theta_2 & 0 & -\sin\theta_2 \\ 0 & 1 & 0 \\ \sin\theta_2 & 0 & \cos\theta_2 \end{pmatrix} \begin{pmatrix} \cos\theta_3 & -\sin\theta_3 & 0 \\ \sin\theta_3 & \cos\theta_3 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.4}$$

The stress as a second-order tensor will be transformed according to the follow formula

$$\begin{pmatrix} \sigma'_{11} & \sigma'_{12} & \sigma'_{13} \\ \sigma'_{21} & \sigma'_{22} & \sigma'_{23} \\ \sigma'_{31} & \sigma'_{32} & \sigma'_{33} \end{pmatrix} = \beta \begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{pmatrix} \beta^T \tag{3.5}$$

where σ'_{ij} denotes the stress tensor in the beam coordinate system and σ_{ij} denotes the stress tensor in the material coordinate system. In engineering notation, the transformation rule can be expressed as

$$\sigma' = T_\sigma \sigma \tag{3.6}$$

with

$$
T_\sigma = \begin{pmatrix}
\beta_{11}^2 & \beta_{12}^2 & \beta_{13}^2 & 2\beta_{12}\beta_{13} & 2\beta_{11}\beta_{13} & 2\beta_{11}\beta_{12} \\
\beta_{11}\beta_{21} & \beta_{12}\beta_{22} & \beta_{13}\beta_{23} & \beta_{13}\beta_{22} + \beta_{12}\beta_{23} & \beta_{13}\beta_{21} + \beta_{11}\beta_{23} & \beta_{12}\beta_{21} + \beta_{11}\beta_{32} \\
\beta_{11}\beta_{31} & \beta_{12}\beta_{32} & \beta_{13}\beta_{33} & \beta_{13}\beta_{32} + \beta_{12}\beta_{33} & \beta_{13}\beta_{31} + \beta_{11}\beta_{33} & \beta_{12}\beta_{31} + \beta_{11}\beta_{32} \\
\beta_{21}^2 & \beta_{22}^2 & \beta_{23}^2 & 2\beta_{22}\beta_{23} & 2\beta_{21}\beta_{23} & 2\beta_{21}\beta_{22} \\
\beta_{21}\beta_{31} & \beta_{22}\beta_{32} & \beta_{23}\beta_{33} & \beta_{23}\beta_{32} + \beta_{22}\beta_{33} & \beta_{23}\beta_{31} + \beta_{21}\beta_{33} & \beta_{22}\beta_{31} + \beta_{21}\beta_{32} \\
\beta_{31}^2 & \beta_{32}^2 & \beta_{33}^2 & 2\beta_{32}\beta_{33} & 2\beta_{31}\beta_{33} & 2\beta_{31}\beta_{32}
\end{pmatrix} \tag{3.7}
$$

It can be derived that the stiffness matrix will be transformed according to the following rule

$$C^* = T_\sigma C T_\sigma^T$$

Using this transformation rule, we can obtain the material properties of each layer expressed in the beam coordinate system, then we average these properties with respect to the thickness of each layer and use this new material property as input for VABS to perform the cross-sectional analysis. Finally, the new Timoshenko stiffness matrix using smeared properties can be obtained as

$$
\begin{bmatrix}
4.076 \times 10^6 & 0 & 0 & 0 & -1.012 \times 10^{-4} & -6.237 \times 10^{-1} \\
 & 8.944 \times 10^5 & -6.475 \times 10^{-1} & -1.570 \times 10^{-2} & 0 & 0 \\
 & & 1.539 \times 10^4 & 1.469 \times 10^{-1} & 0 & 0 \\
 & & & 1.491 \times 10^0 & 0 & 0 \\
 & \mathit{symmetry} & & & 1.475 \times 10^0 & 5.945 \times 10^{-6} \\
 & & & & & 2.081 \times 10^3
\end{bmatrix}
$$

The difference of the major stiffness terms using smeared properties is compared to the original direct analysis with modeling all the layered details is tabulated in Table 3.1. From this table, it is obviously to see the difference between modeling layered details and using smeared properties. To be consistent, I use the same mesh input, include the same element size, same element type and the same fineness. Clearly, using smeared properties will

Table 3.1: Results comparison

	cross-sectional property	smeared property	% Diff.(Origin)
EA	4.108E+06	4.056E+06	1.3
GJ	1.523E+00	1.475E+00	3.2
EI_{22}	1.509E+00	1.475E+00	2.3
EI_{33}	2.173E+03	2.081E+03	4.2
S_{22}	9.909E+05	8.944E+05	9.7
S_{33}	1.488E+04	1.539E+04	3.4
Mesh time	1.2 seconds	0.1 seconds	1100

achieve simpler modeling and possibly faster computation. However, the loss of accuracy, ranging from 1% to 9% for this case, is not negligible. It is expected that for more complex structures, such as realistic composite rotor blades, the loss of accuracy will be more significant and same is true with the gain of modeling. However, with the current efficiency of VABS and convenience of modeling layered details using one of the preprocessors (VABS-IDE, VABS-GUI, and PreVABS), the gain of efficiency might not be justified for the loss of accuracy.

3.2 Mesh Convergence

Mesh convergence studies the convergence of results with respect to the increasing number of elements in a finite element mesh. It is well-known fact that finite element solutions depends on the number of elements used in a mesh. However, the solution should not change significantly after the mesh reaches certainly less of fineness. For example, If we do the first order elements, a quick way to check mesh convergence is to use the program to convert the elements to a second order and then check whether the solution changes. If it does not change significantly, then the first-order mesh is converged. Let us still use the simple composite strip to study mesh convergence of VABS, which is a question raised by one of VABS users in industry.

First, let us mesh each layer using a 8-noded quadrilateral element (Table 3.2).

Table 3.2: Mesh data 1

Heal toler	Max size	Min size	Fineness	Mesh time	Element type	Element No.
0.01	0.08	0.002	User Defined	0.4S	QUAD8	8

Table 3.3: Mesh data 2

Heal toler	Max size	Min size	Fineness	Mesh time	Element type	Element No.
0.01	0.004	0.002	User Defined	2S	QUAD8	160

The corresponding stiffness result is shown below:

$$\begin{bmatrix} 4.101\times10^{6} & 4.466\times10^{3} & -9.168\times10^{0} & -1.091\times10^{3} & -1.627\times10^{2} & -1.589\times10^{2} \\ & 9.814\times10^{5} & 1.265\times10^{1} & -1.007\times10^{2} & 6.002\times10^{2} & 1.612\times10^{-1} \\ & & 7.972\times10^{3} & 1.922\times10^{-1} & -2.456\times10^{-1} & 5.169\times10^{-1} \\ & & & 1.505\times10^{0} & -3.766\times10^{-2} & 1.390\times10^{-1} \\ & symmetry & & & 1.479\times10^{0} & 5.245\times10^{-2} \\ & & & & & 2.156\times10^{3} \end{bmatrix} \tag{3.8}$$

If we start to refine the mesh, and use the control parameters in Table 3.3, we can create a mesh with 160 8-noded quadrilateral elements. The corresponding stiffness results are as follows:

$$\begin{bmatrix} 4.102\times10^{6} & 4.479\times10^{3} & -9.168\times10^{0} & -1.091\times10^{3} & -1.627\times10^{2} & -1.589\times10^{2} \\ & 9.818\times10^{5} & 1.268\times10^{1} & -1.011\times10^{2} & 6.014\times10^{2} & 1.631\times10^{-1} \\ & & 7.965\times10^{3} & 1.923\times10^{-1} & -2.467\times10^{-1} & 5.173\times10^{-1} \\ & & & 1.513\times10^{0} & -3.767\times10^{-2} & 1.410\times10^{-1} \\ & symmetry & & & 1.485\times10^{0} & 5.246\times10^{-2} \\ & & & & & 2.164\times10^{3} \end{bmatrix} \tag{3.9}$$

We continue to refine the mesh according to Table 3.4, and obtain the following stiffness matrix

Table 3.4: Mesh data 3

Heal toler	Max size	Min size	Fineness	Mesh time	Element type	Element No.
0.01	0.001	0.001	User Define	12.1S	QUAD8	1280

Table 3.5: Mesh data 4

Heal toler	Max size	Min size	Fineness	Mesh time	Element type	Element No.
0.01	0.0004	0.0002	User Define	40.1S	QUAD8	16000

$$\begin{bmatrix} 4.103\times10^{6} & 8.131\times10^{3} & -4.267\times10^{1} & -1.095\times10^{3} & -1.631\times10^{2} & -1.624\times10^{2} \\ & 9.820\times10^{5} & -2.298\times10^{2} & -9.522\times10^{2} & 5.903\times10^{2} & 4.850\times10^{1} \\ & & 7.923\times10^{3} & -5.388\times10^{0} & -8.556\times10^{-1} & 1.412\times10^{0} \\ & & & 1.515\times10^{0} & -3.221\times10^{-2} & 1.346\times10^{-1} \\ & symmetry & & & 1.485\times10^{0} & 5.018\times10^{-2} \\ & & & & & 2.166\times10^{3} \end{bmatrix} \tag{3.10}$$

More refinements are also carried out. According to the mesh data in Table 3.5, we obtain the following stiffness matrix:

$$\begin{bmatrix} 4.105\times10^{6} & 8.124\times10^{3} & -4.278\times10^{1} & -1.095\times10^{3} & -1.631\times10^{2} & -1.624\times10^{2} \\ & 9.820\times10^{5} & -2.305\times10^{2} & -9.531\times10^{2} & 5.903\times10^{2} & 4.850\times10^{1} \\ & & 7.846\times10^{3} & -5.392\times10^{0} & -8.577\times10^{-1} & 1.406\times10^{0} \\ & & & 1.517\times10^{0} & -4.221\times10^{-2} & 1.346\times10^{-1} \\ & symmetry & & & 1.486\times10^{0} & 9.098\times10^{-2} \\ & & & & & 2.170\times10^{3} \end{bmatrix} \tag{3.11}$$

Refine the mesh according to Table 3.6, we obtain the following Timoshenko stiffness matrix:

Table 3.6: Mesh data 5

Heal toler	Max size	Min size	Fineness	Mesh time	Element type	Element No.
0.01	0.0001	0.0001	User Define	70.1S	QUAD8	35990

Table 3.7: Mesh data 6

Heal toler	Max size	Min size	Fineness	Mesh time	Element type	Element No.
0.001	0.00008	0.00008	User Define	120.2S	QUAD8	56800

$$\begin{bmatrix} 4.107\times10^{6} & 1.133\times10^{4} & 4.107\times10^{0} & -1.094\times10^{3} & -1.647\times10^{2} & 4.082\times10^{1} \\ & 9.853\times10^{5} & -2.884\times10^{1} & -9.075\times10^{1} & 6.001\times10^{2} & -1.412\times10^{1} \\ & & 7.761\times10^{3} & -2.113\times10^{-1} & 3.439\times10^{-2} & 1.084\times10^{0} \\ & & & 1.520\times10^{0} & -3.709\times10^{-2} & -2.421\times10^{-2} \\ & \textit{symmetry} & & & 1.486\times10^{0} & 1.128\times10^{-2} \\ & & & & & 2.172\times10^{3} \end{bmatrix} \tag{3.12}$$

According to the mesh refinements in Table 3.7, we obtain the following stiffness matrix:

$$\begin{bmatrix} 4.108\times10^{6} & 1.133\times10^{4} & 4.107\times10^{0} & -1.094\times10^{3} & -1.647\times10^{2} & 4.082\times10^{1} \\ & 9.854\times10^{5} & -2.884\times10^{1} & -9.075\times10^{1} & 6.001\times10^{2} & -1.412\times10^{1} \\ & & 7.678\times10^{3} & -2.113\times10^{-1} & 3.439\times10^{-2} & 1.084\times10^{0} \\ & & & 1.521\times10^{0} & -3.709\times10^{-2} & -2.421\times10^{-2} \\ & \textit{symmetry} & & & 1.487\times10^{0} & 1.128\times10^{-2} \\ & & & & & 2.172\times10^{3} \end{bmatrix} \tag{3.13}$$

The last refinement of the mesh were performed according to Table 3.8. The stiffness matrix is:

Table 3.8: Mesh data 7

Heal toler	Max size	Min size	Fineness	Mesh time	Element type	Element No.
0.001	0.00005	0.00005	Very fine	240.6S	QUAD8	96000

	data 1	data 2	data 3	data 4	data 5	data 6	data7
EA	4.101E6	4.102E6	4.103E6	4.105E6	4.107E6	4.108E6	4.108E6
GJ	1.505	1.513	1.515	1.517	1.521	1.521	1.523
EI_{22}	1.479	1.485	1.486	1.486	1.486	1.487	1.487
EI_{33}	2.156E3	2.164E3	2.166E3	2.170E3	2.172E3	2.172E3	2.172E3
S_{22}	9.814E5	9.818E5	9.820E5	9.820E5	9.853E5	9.854E5	9.854E5
S_{33}	7.972E3	7.965E3	7.923E3	7.846E3	7.761E3	7.678E3	7.678E3

$$\begin{bmatrix} 4.108\times10^{6} & 1.133\times10^{4} & 4.107\times10^{0} & -1.094\times10^{3} & -1.647\times10^{2} & 4.082\times10^{1} \\ & 9.854\times10^{5} & -2.884\times10^{1} & -9.075\times10^{1} & 6.001\times10^{2} & -1.412\times10^{1} \\ & & 7.678\times10^{3} & -2.113\times10^{-1} & 3.439\times10^{-2} & 1.084\times10^{0} \\ & & & 1.523\times10^{0} & -3.709\times10^{-2} & -2.421\times10^{-2} \\ & \mathit{symmetry} & & & 1.487\times10^{0} & 1.128\times10^{-2} \\ & & & & & 2.172\times10^{3} \end{bmatrix} \tag{3.14}$$

If we neglect the small off-diagonal terms, we tabulate the diagonal terms to check the convergence in Table 3.9.

From the above table, it apparently shows that the results are converging. As the number of elements increases, the better accuracy of results can be obtained. However, when more refined mesh is used, the mesh time and analysis time will also increase, sometimes dramatically. Due to condition of my own computer, it is slow. so, the mesh time I listed above may changed depend on different computers. However, the tendency, which changing element number will lead to increasing mesh time will remain true. Taking EA for example, the convergence can be demonstrated graphically in Fig. 3.3:

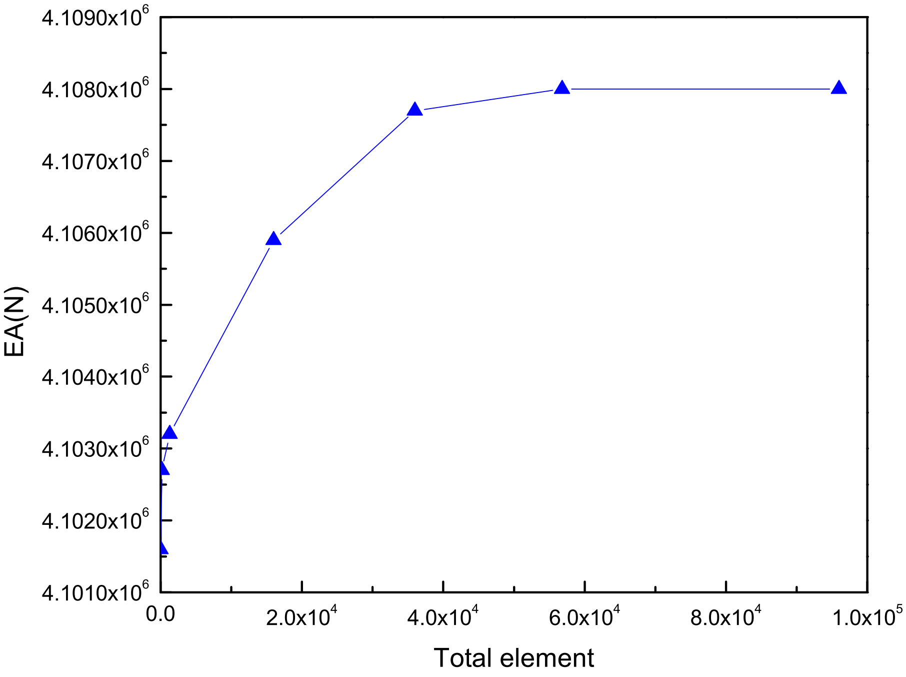


Fig. 3.3: Mesh Convergence.

Chapter 4

Results and Discussions

In Chapter 2, we modeled four different cross-sections using VABS-IDE and VABS-GUI. From these results, I found that the simpler the cross-section, the more accurate results can be obtained, especially for the VABS-IDE since the element size of VABS-IDE is hard to control when I generate the mesh. Furthermore, there are bugs that will occur when I build the geometry and I believe, in comparison, that VABS-GUI is much better at meshing than VABS-IDE.

Example 1 is a very simple cross-section, the way to set up the geometry and mesh the model is almost the same. As a result, the cross-sectional properties calculated by both VABS-IDE and VABS-GUI are all the same and at the same time these two results perfectly match the linear elasticity theory. Nevertheless, example discloses the fact that one should try to respect the symmetry of the cross-section which create the finite element mesh. VABS-GUI is much better in this aspect.

Referring to example 2, it is an isotropic channel section, which means its shape is not symmetric. This cross-section is more complex than the first example, however, it's still easy to do the geometry and mesh. Therefore, the cross-sectional properties calculated by both VABS-IDE and VABS-GUI are still close to each other. And in this example, VABS-IDE and VABS-GUI shows advantage over PreVABS as the section can be modeled directly without adding any dummy materials. Example 3 is a multilayer composite pipe. For this cross-section it has two different layers with the same material properties but with the different fiber orientation. The difference between different codes gets bigger particularly for one of the transverse shear stiffness. It is suggested to look at the PreVABS model more closely to find out the cause of this discrepancy.

In the fourth example, we encounter an isotropic blade-like section. In this example, we disclosed a strange bug of VABS-IDE that meshing will change the original geometry. That is also the main reason cause the noticeable difference between VABS-IDE and VABS-GUI.

In Chapter 3, we used VABS-IDE to investigate the loss of accuracy due to using smeared properties for composite beam modeling. We find out even for a very simple composite strip, using smeared properties will cause significant loss of accuracy although it will improve the efficiency in meshing and analysis. Lastly, we also studied the mesh convergence of VABS results, which shows a similar trend as any codes based on the finite element technique.

4.1 Comparison

Because of using the same analysis code of VABS, the way to solve problems by both software is the same. However, we still get some slight differences for most of the cases within reasonable range. These differences are caused by the preprocessor including geometry creating and meshing. As far as the meshing is concerned, VABS-IDE and VABS-GUI are dramatically different. For VABS-IDE, we have four choices to do the mesh, besides, we can set the element size in order to refine the mesh. Referring to VABS-GUI, mesh refinement can be done easily using the function bottom called “Refine by splitting”. By doing this way, program can automatically refine the mesh. I would like to present some advantages and disadvantages of both VABS-IDE and VABS-GUI, although they all based on the same VABS code. However, the different interface makes them have their own core competitiveness.

4.1.1 VABS-IDE

Distinguished interface makes VABS-IDE easy to handling. Because of convenient tool bars on the both top and right side of the interface, even you are the first time to use it, it’s not hard for you to find any functional bottom. Message box and parameter control box are provided, which means you can easily see the mesh and the solution results. For the mesh part, you can choose four different elements types: 3 nodes triangle, 6 nodes triangle, 4 nodes quadratic and 8 nodes quadratic, which can meet your different requirement of accuracy.

Furthermore, you can also choose different size of each element. However, VABS-IDE still exist some disadvantages that may affect the modeling procedure. Firstly, because of the existence of message box and parameter control box take over too much space of interface, the geometry panel stays relatively small, especially, when I did one complex model (such as multiple layers, plenty of nodes and etc.), the remaining space for the whole cross-section is not enough. Then, it's not easy for VABS-IDE to make a change in input file; every time when I made a wrong input (such as points, straight lines and arcs.), I have to delete what I just input. This implies, if we input a wrong number while we are modeling a complex cross-section, it's huge work of re-modeling. Till now, VABS-IDE is a commercial software, thus, this software has a professional technical support team.

4.1.2 VABS-GUI

We can do the modeling in VABS-GUI directly on the huge interface, easily to input any parameters. Especially, VABS-GUI can generate a txt file to edit geometry, material properties and layup information. You can make any reasonable changes in this txt file, then if you press the 'RELOAD' bottom, the newly geometry shape will be shown at the interface. In VABS-GUI, you can easily pick any point, line, face that you want to choose. Referring to the meshing part, VABS-GUI provides many kind of ways to make mesh and a convenient way to refine the mesh. Solver includes both Constitutive Modeling and Recovery. VABS-GUI also includes a postprocessor, which is convenient for visualizing the results. On the other hand, when I was using VABS-GUI to do modeling, I found some aspects, which are not so convenient to handling. First of all, all the tools are inside of a pull-down menu, it looks clean and tidy, but it's easily to choose a wrong bottom. Then, VABS-GUI is not that visualized to see modeling, mesh, and result information directly inside of interface. Instead, we have to open other files to check the geometry and result. For this part, I will suggest to add one margin on the left side as a message box, maybe it will reduce the huge interface, however, it will become much easier to do the modeling in the interface directly. Furthermore, the default way to save mesh file is a temp file in system disk. As a result, every time I finished the meshing and save it, when I started to solve, there will have an error message came out, saying that “Please Mesh Your Model and SAVE”. Then I have to find that temp file and copy one mesh file, then paste it in the root file of VABS-GUI. Furthermore, if we use orthotropic materials, after adding the material properties, it will have a crush down error, when we run the solver. Lastly, about the “layup information”, in the lay-up definition box, I can only define lay-up angles for 8 layers, which means the current version cannot model a more complex section with more than 8 layers.

4.2 Recommendation

To sum up, although VABS-GUI is currently in test version, there is still long way to debug in order to make it perfect and improve; however, we can already see the convenient, effective, accurate modeling. As a suggestion, I would say the VABS-GUI should have more functional bottom on the interface. What’s more, some bugs still need to be fixed, including program crush down, reading the geometry file only with the name of ”VABSGUI.geo”, and limitation of 8 layers layup information in the control panel.

Referring to the VABS-IDE, as a commercial software, it is definitely powerful and easy to handle. However, there are two aspects need to be considered. First one is the controllable element size when one does the meshing. Another one is the geometry part. The object browser is too small to make any necessarily changes when we encounter some multilayers and complex geometry.

References

[1] Chen, H., Yu, W., and Capellaro, M., 2010. “A critical assessment of computer tools for calculating composite wind turbine blade properties.” *Wind Energy,* **13**(6), pp. 497–516.

[2] Yu, W., 2002. *Variational asymptotic modeling of composite dimensionally reducible structures.*

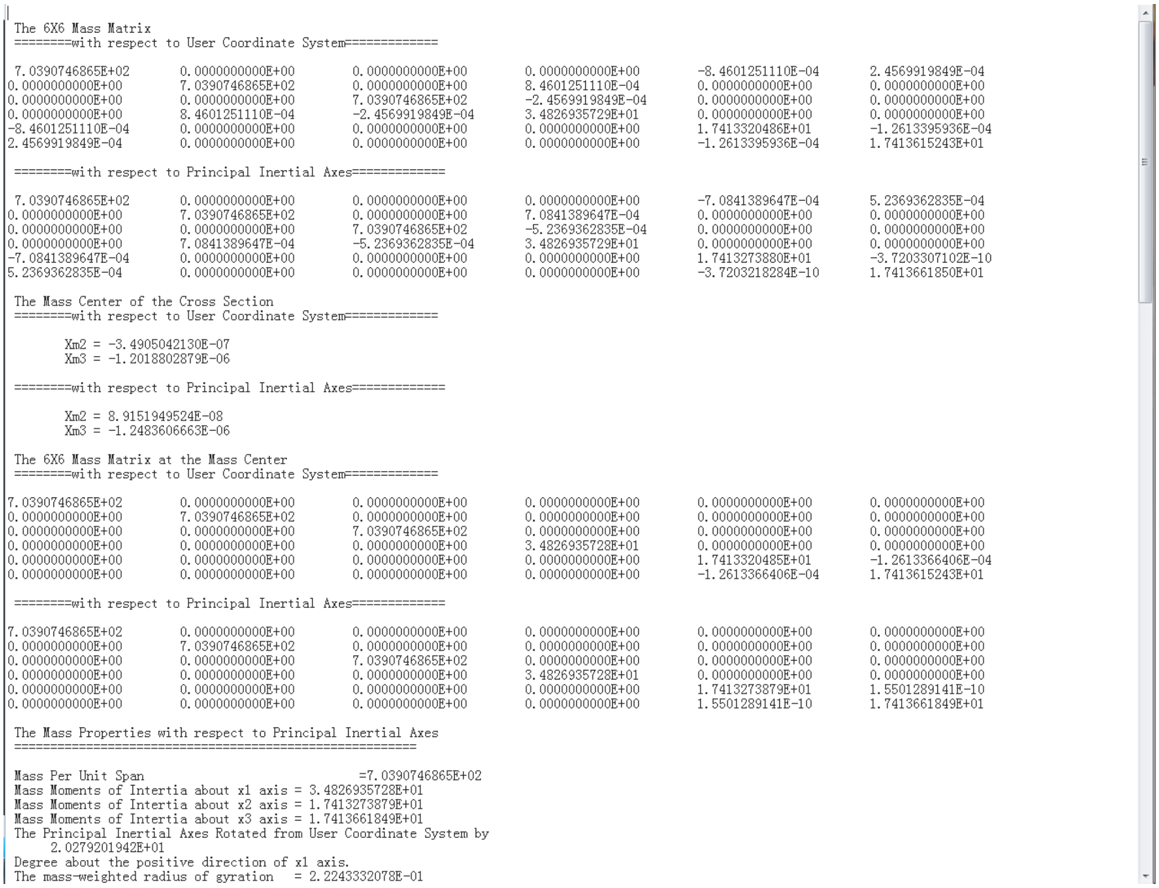
[3] Yu, W., Ho, J. C., and Hodges, D. H., 2012. “Variational asymptotic beam sectional analysis - an updated version.” *International Journal of Engineering Science.* in press.

[4] Geuzaine, C., and Remacle, J.-F., 2012. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. Tech. Rep. http://geuz.org/gmsh/.

Appendices

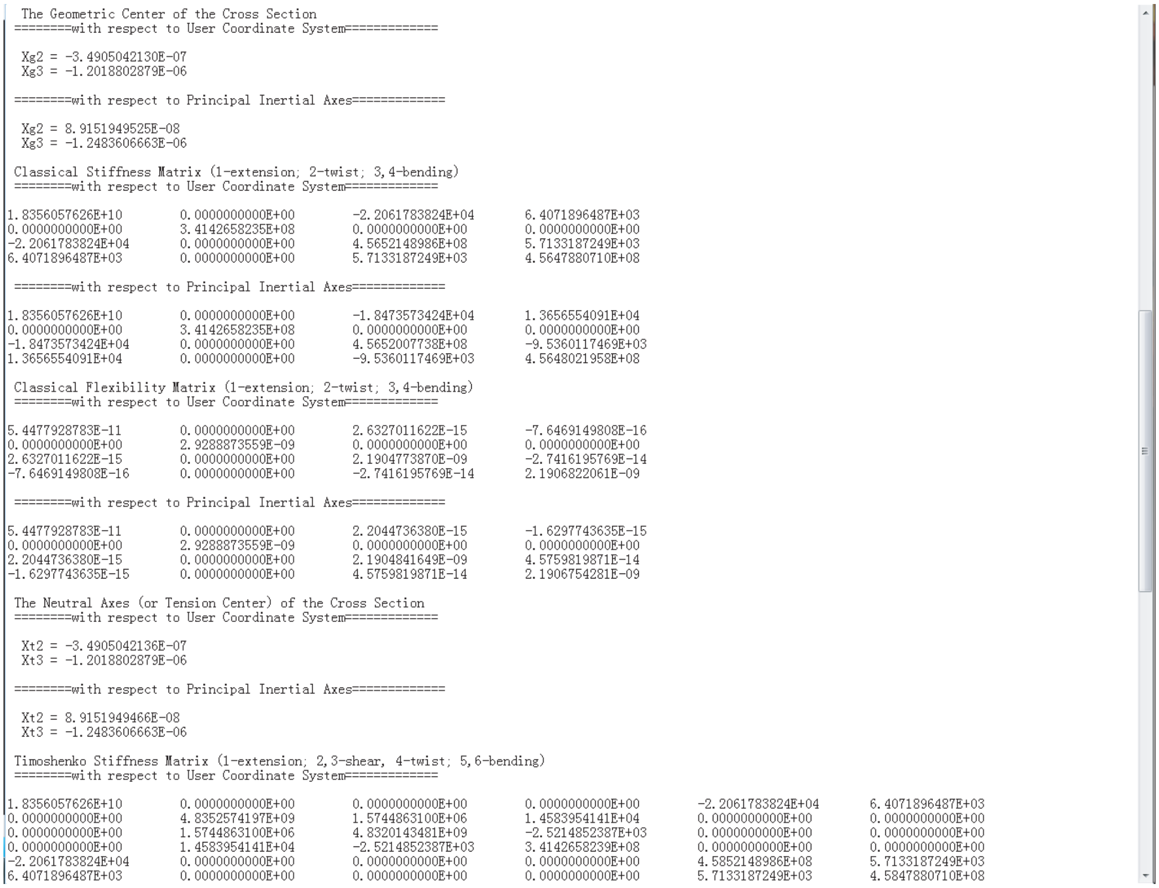
Appendix A

The output file by VABS-IDE



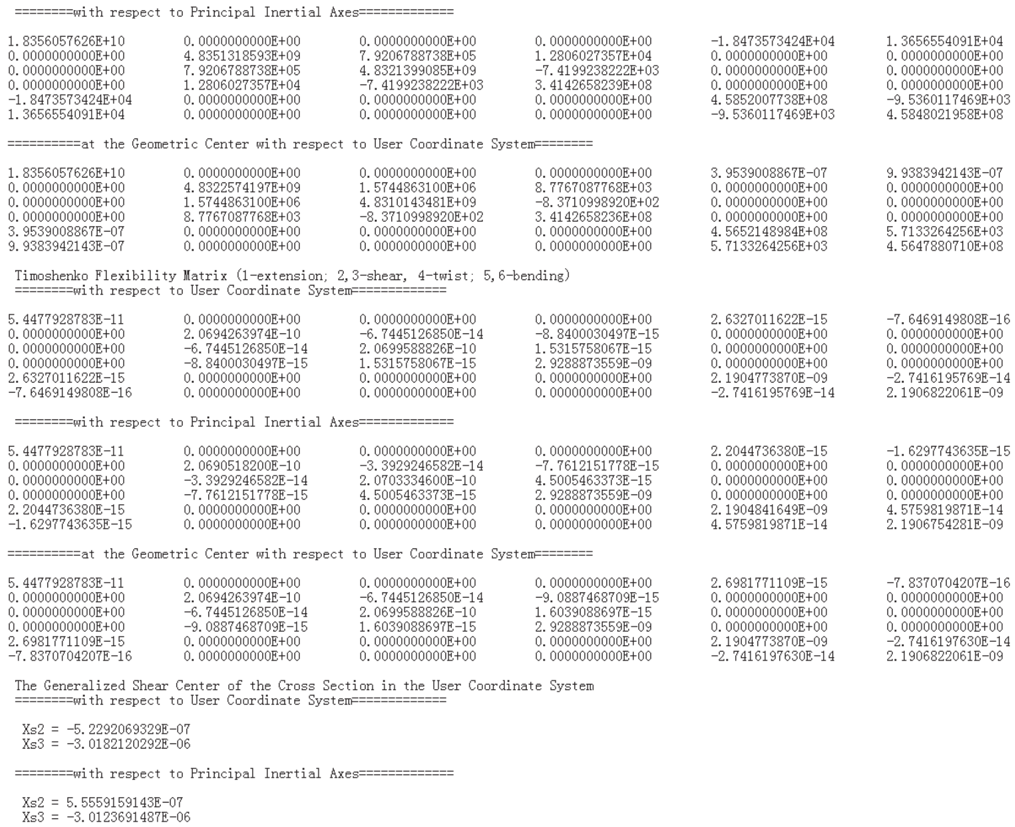
```
|
 The 6X6 Mass Matrix
 ========with respect to User Coordinate System=============

 7.0390746865E+02       0.0000000000E+00        0.0000000000E+00        0.0000000000E+00        -8.4601251110E-04       2.4569919849E-04
0.0000000000E+00        7.0390746865E+02        0.0000000000E+00        8.4601251110E-04        0.0000000000E+00        0.0000000000E+00
0.0000000000E+00        0.0000000000E+00        7.0390746865E+02        -2.4569919849E-04       0.0000000000E+00        0.0000000000E+00
0.0000000000E+00        8.4601251110E-04        -2.4569919849E-04       3.4826935729E+01        0.0000000000E+00        0.0000000000E+00
-8.4601251110E-04       0.0000000000E+00        0.0000000000E+00        0.0000000000E+00        1.7413320486E+01        -1.2613395936E-04
2.4569919849E-04        0.0000000000E+00        0.0000000000E+00        0.0000000000E+00        -1.2613395936E-04       1.7413615243E+01

 ========with respect to Principal Inertial Axes=============

 7.0390746865E+02       0.0000000000E+00        0.0000000000E+00        0.0000000000E+00        -7.0841389647E-04       5.2369362835E-04
0.0000000000E+00        7.0390746865E+02        0.0000000000E+00        7.0841389647E-04        0.0000000000E+00        0.0000000000E+00
0.0000000000E+00        0.0000000000E+00        7.0390746865E+02        -5.2369362835E-04       0.0000000000E+00        0.0000000000E+00
0.0000000000E+00        7.0841389647E-04        -5.2369362835E-04       3.4826935729E+01        0.0000000000E+00        0.0000000000E+00
-7.0841389647E-04       0.0000000000E+00        0.0000000000E+00        0.0000000000E+00        1.7413273880E+01        -3.7203307102E-10
5.2369362835E-04        0.0000000000E+00        0.0000000000E+00        0.0000000000E+00        -3.7203218284E-10       1.7413661850E+01

 The Mass Center of the Cross Section
 ========with respect to User Coordinate System=============

        Xm2 = -3.4905042130E-07
        Xm3 = -1.2018802879E-06

 ========with respect to Principal Inertial Axes=============

        Xm2 = 8.9151949524E-08
        Xm3 = -1.2483606663E-06

 The 6X6 Mass Matrix at the Mass Center
 ========with respect to User Coordinate System=============

7.0390746865E+02        0.0000000000E+00        0.0000000000E+00        0.0000000000E+00        0.0000000000E+00        0.0000000000E+00
0.0000000000E+00        7.0390746865E+02        0.0000000000E+00        0.0000000000E+00        0.0000000000E+00        0.0000000000E+00
0.0000000000E+00        0.0000000000E+00        7.0390746865E+02        0.0000000000E+00        0.0000000000E+00        0.0000000000E+00
0.0000000000E+00        0.0000000000E+00        0.0000000000E+00        3.4826935728E+01        0.0000000000E+00        0.0000000000E+00
0.0000000000E+00        0.0000000000E+00        0.0000000000E+00        0.0000000000E+00        1.7413320485E+01        -1.2613366406E-04
0.0000000000E+00        0.0000000000E+00        0.0000000000E+00        0.0000000000E+00        -1.2613366406E-04       1.7413615243E+01

 ========with respect to Principal Inertial Axes=============

7.0390746865E+02        0.0000000000E+00        0.0000000000E+00        0.0000000000E+00        0.0000000000E+00        0.0000000000E+00
0.0000000000E+00        7.0390746865E+02        0.0000000000E+00        0.0000000000E+00        0.0000000000E+00        0.0000000000E+00
0.0000000000E+00        0.0000000000E+00        7.0390746865E+02        0.0000000000E+00        0.0000000000E+00        0.0000000000E+00
0.0000000000E+00        0.0000000000E+00        0.0000000000E+00        3.4826935728E+01        0.0000000000E+00        0.0000000000E+00
0.0000000000E+00        0.0000000000E+00        0.0000000000E+00        0.0000000000E+00        1.7413273879E+01        1.5501289141E-10
0.0000000000E+00        0.0000000000E+00        0.0000000000E+00        0.0000000000E+00        1.5501289141E-10        1.7413661849E+01

 The Mass Properties with respect to Principal Inertial Axes
 ========================================================

 Mass Per Unit Span                               =7.0390746865E+02
 Mass Moments of Intertia about x1 axis = 3.4826935728E+01
 Mass Moments of Intertia about x2 axis = 1.7413273879E+01
 Mass Moments of Intertia about x3 axis = 1.7413661849E+01
 The Principal Inertial Axes Rotated from User Coordinate System by
      2.0279201942E+01
 Degree about the positive direction of x1 axis.
 The mass-weighted radius of gyration   = 2.2243332078E-01
```

Fig. A.1: The first part of one output file by VABS-IDE of example 1



```
 The Geometric Center of the Cross Section
========with respect to User Coordinate System============

 Xg2 = -3.4905042130E-07
 Xg3 = -1.2018802879E-06

========with respect to Principal Inertial Axes============

 Xg2 = 8.9151949525E-08
 Xg3 = -1.2483606663E-06

Classical Stiffness Matrix (1-extension; 2-twist; 3,4-bending)
========with respect to User Coordinate System============

1.8356057626E+10       0.0000000000E+00       -2.2061783824E+04      6.4071896487E+03
0.0000000000E+00       3.4142658235E+08       0.0000000000E+00       0.0000000000E+00
-2.2061783824E+04      0.0000000000E+00       4.5652148986E+08       5.7133187249E+03
6.4071896487E+03       0.0000000000E+00       5.7133187249E+03       4.5647880710E+08

========with respect to Principal Inertial Axes============

1.8356057626E+10       0.0000000000E+00       -1.8473573424E+04      1.3656554091E+04
0.0000000000E+00       3.4142658235E+08       0.0000000000E+00       0.0000000000E+00
-1.8473573424E+04      0.0000000000E+00       4.5652007738E+08       -9.5360117469E+03
1.3656554091E+04       0.0000000000E+00       -9.5360117469E+03      4.5648021958E+08

Classical Flexibility Matrix (1-extension; 2-twist; 3,4-bending)
========with respect to User Coordinate System============

5.4477928783E-11       0.0000000000E+00       2.6327011622E-15       -7.6469149808E-16
0.0000000000E+00       2.9288873559E-09       0.0000000000E+00       0.0000000000E+00
2.6327011622E-15       0.0000000000E+00       2.1904773870E-09       -2.7416195769E-14
-7.6469149808E-16      0.0000000000E+00       -2.7416195769E-14      2.1906822061E-09

========with respect to Principal Inertial Axes============

5.4477928783E-11       0.0000000000E+00       2.2044736380E-15       -1.6297743635E-15
0.0000000000E+00       2.9288873559E-09       0.0000000000E+00       0.0000000000E+00
2.2044736380E-15       0.0000000000E+00       2.1904841649E-09       4.5759819871E-14
-1.6297743635E-15      0.0000000000E+00       4.5759819871E-14       2.1906754281E-09

The Neutral Axes (or Tension Center) of the Cross Section
========with respect to User Coordinate System============

 Xt2 = -3.4905042136E-07
 Xt3 = -1.2018802879E-06

========with respect to Principal Inertial Axes============

 Xt2 = 8.9151949466E-08
 Xt3 = -1.2483606663E-06

Timoshenko Stiffness Matrix (1-extension; 2,3-shear, 4-twist; 5,6-bending)
========with respect to User Coordinate System============

1.8356057626E+10       0.0000000000E+00       0.0000000000E+00       0.0000000000E+00       -2.2061783824E+04      6.4071896487E+03
0.0000000000E+00       4.8352574197E+09       1.5744863100E+06       1.4583954141E+04       0.0000000000E+00       0.0000000000E+00
0.0000000000E+00       1.5744863100E+06       4.8320143481E+09       -2.5214852387E+03      0.0000000000E+00       0.0000000000E+00
0.0000000000E+00       1.4583954141E+04       -2.5214852387E+03      3.4142658239E+08       0.0000000000E+00       0.0000000000E+00
-2.2061783824E+04      0.0000000000E+00       0.0000000000E+00       0.0000000000E+00       4.5852148986E+08       5.7133187249E+03
6.4071896487E+03       0.0000000000E+00       0.0000000000E+00       0.0000000000E+00       5.7133187249E+03       4.5847880710E+08
```

Fig. A.2: The second part of one output file by VABS-IDE of example 1

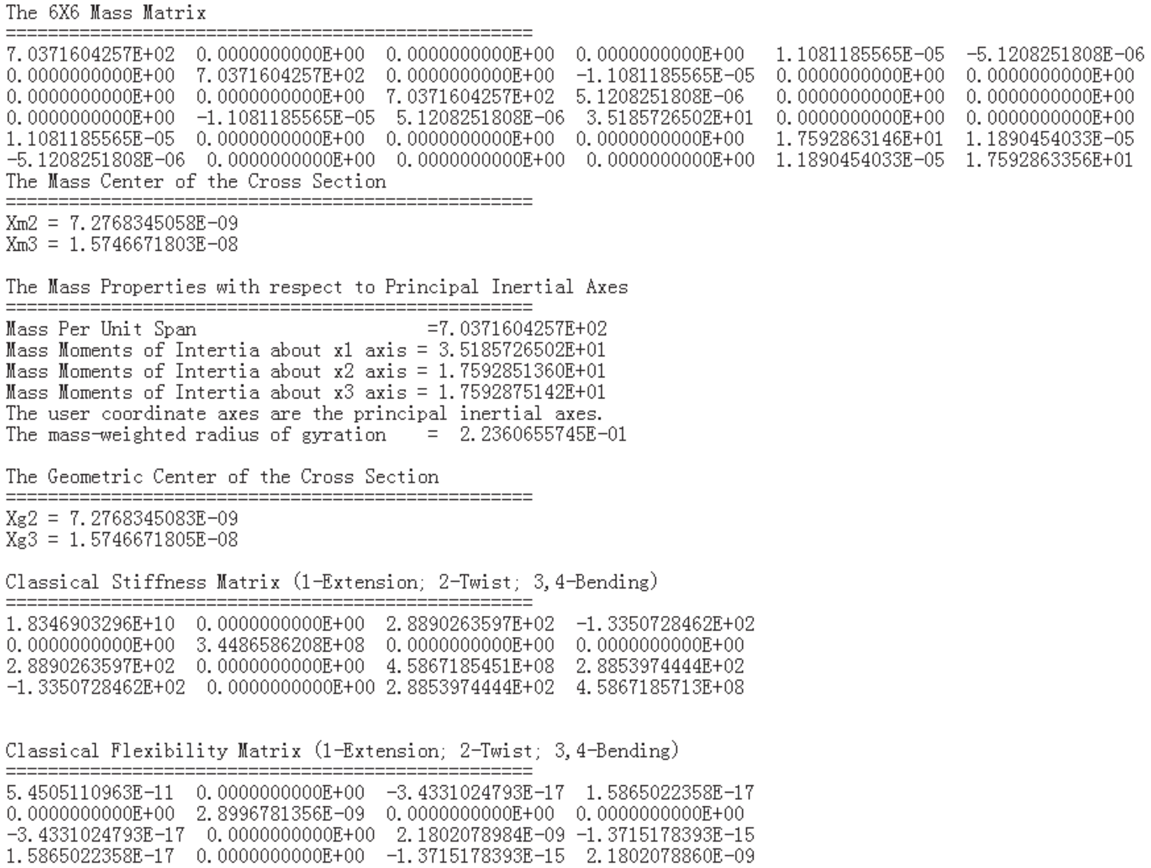


```
 ========with respect to Principal Inertial Axes=============

1.8356057626E+10        0.0000000000E+00        0.0000000000E+00        0.0000000000E+00        -1.8473573424E+04       1.3656554091E+04
0.0000000000E+00        4.8351318593E+09        7.9206788738E+05        1.2806027357E+04        0.0000000000E+00        0.0000000000E+00
0.0000000000E+00        7.9206788738E+05        4.8321399085E+09        -7.4199238222E+03       0.0000000000E+00        0.0000000000E+00
0.0000000000E+00        1.2806027357E+04        -7.4199238222E+03       3.4142658239E+08        0.0000000000E+00        0.0000000000E+00
-1.8473573424E+04       0.0000000000E+00        0.0000000000E+00        0.0000000000E+00        4.5852007738E+08        -9.5360117469E+03
1.3656554091E+04        0.0000000000E+00        0.0000000000E+00        0.0000000000E+00        -9.5360117469E+03       4.5848021958E+08

==========at the Geometric Center with respect to User Coordinate System========

1.8356057626E+10        0.0000000000E+00        0.0000000000E+00        0.0000000000E+00        3.9539008867E-07        9.9383942143E-07
0.0000000000E+00        4.8322574197E+09        1.5744863100E+06        8.7767087768E+03        0.0000000000E+00        0.0000000000E+00
0.0000000000E+00        1.5744863100E+06        4.8310143481E+09        -8.3710998920E+02       0.0000000000E+00        0.0000000000E+00
0.0000000000E+00        8.7767087768E+03        -8.3710998920E+02       3.4142658236E+08        0.0000000000E+00        0.0000000000E+00
3.9539008867E-07        0.0000000000E+00        0.0000000000E+00        0.0000000000E+00        4.5652148984E+08        5.7133264256E+03
9.9383942143E-07        0.0000000000E+00        0.0000000000E+00        0.0000000000E+00        5.7133264256E+03        4.5647880710E+08

 Timoshenko Flexibility Matrix (1-extension; 2,3-shear, 4-twist; 5,6-bending)
 ========with respect to User Coordinate System=============

5.4477928783E-11        0.0000000000E+00        0.0000000000E+00        0.0000000000E+00        2.6327011622E-15        -7.6469149808E-16
0.0000000000E+00        2.0694263974E-10        -6.7445126850E-14       -8.8400030497E-15       0.0000000000E+00        0.0000000000E+00
0.0000000000E+00        -6.7445126850E-14       2.0699588826E-10        1.5315758067E-15        0.0000000000E+00        0.0000000000E+00
0.0000000000E+00        -8.8400030497E-15       1.5315758067E-15        2.9288873559E-09        0.0000000000E+00        0.0000000000E+00
2.6327011622E-15        0.0000000000E+00        0.0000000000E+00        0.0000000000E+00        2.1904773870E-09        -2.7416195769E-14
-7.6469149808E-16       0.0000000000E+00        0.0000000000E+00        0.0000000000E+00        -2.7416195769E-14       2.1906822061E-09

 ========with respect to Principal Inertial Axes=============

5.4477928783E-11        0.0000000000E+00        0.0000000000E+00        0.0000000000E+00        2.2044736380E-15        -1.6297743635E-15
0.0000000000E+00        2.0690518200E-10        -3.3929246582E-14       -7.7612151778E-15       0.0000000000E+00        0.0000000000E+00
0.0000000000E+00        -3.3929246582E-14       2.0703334600E-10        4.5005463373E-15        0.0000000000E+00        0.0000000000E+00
0.0000000000E+00        -7.7612151778E-15       4.5005463373E-15        2.9288873559E-09        0.0000000000E+00        0.0000000000E+00
2.2044736380E-15        0.0000000000E+00        0.0000000000E+00        0.0000000000E+00        2.1904841649E-09        4.5759819871E-14
-1.6297743635E-15       0.0000000000E+00        0.0000000000E+00        0.0000000000E+00        4.5759819871E-14        2.1906754281E-09

==========at the Geometric Center with respect to User Coordinate System========

5.4477928783E-11        0.0000000000E+00        0.0000000000E+00        0.0000000000E+00        2.6981771109E-15        -7.8370704207E-16
0.0000000000E+00        2.0694263974E-10        -6.7445126850E-14       -9.0887468709E-15       0.0000000000E+00        0.0000000000E+00
0.0000000000E+00        -6.7445126850E-14       2.0699588826E-10        1.6039088697E-15        0.0000000000E+00        0.0000000000E+00
0.0000000000E+00        -9.0887468709E-15       1.6039088697E-15        2.9288873559E-09        0.0000000000E+00        0.0000000000E+00
2.6981771109E-15        0.0000000000E+00        0.0000000000E+00        0.0000000000E+00        2.1904773870E-09        -2.7416197630E-14
-7.8370704207E-16       0.0000000000E+00        0.0000000000E+00        0.0000000000E+00        -2.7416197630E-14       2.1906822061E-09

 The Generalized Shear Center of the Cross Section in the User Coordinate System
 ========with respect to User Coordinate System=============

  Xs2 = -5.2292069329E-07
  Xs3 = -3.0182120292E-06

 ========with respect to Principal Inertial Axes=============

  Xs2 = 5.5559159143E-07
  Xs3 = -3.0123691487E-06
```

Fig. A.3: The third part of one output file by VABS-IDE of example 1

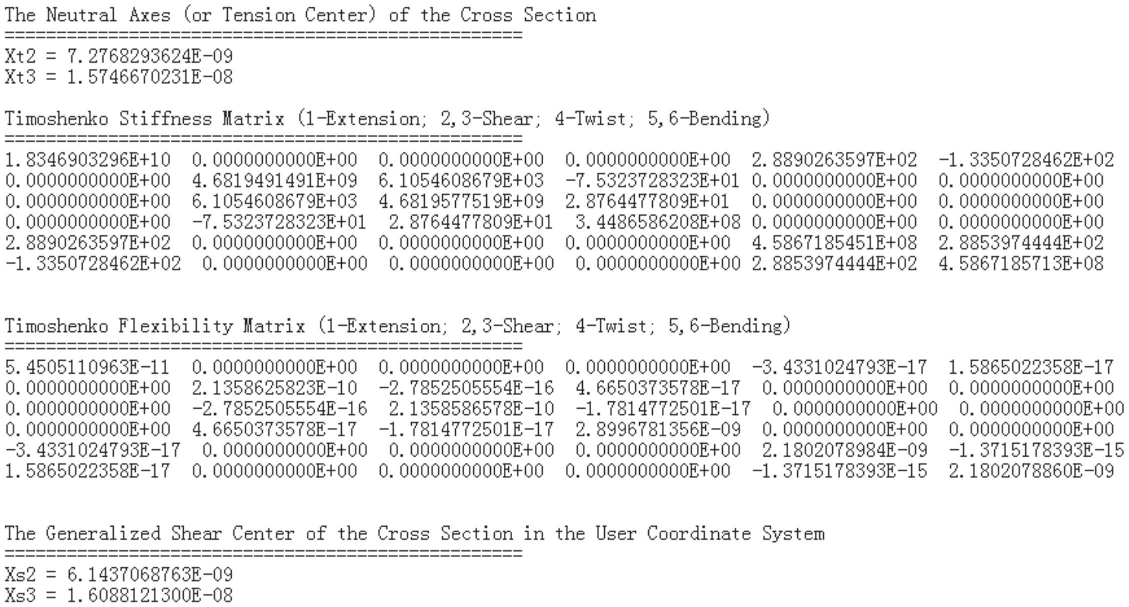
Appendix B

The output file by VABS-GUI



```
The 6X6 Mass Matrix
==================================================
7.0371604257E+02  0.0000000000E+00  0.0000000000E+00  0.0000000000E+00   1.1081185565E-05  -5.1208251808E-06
0.0000000000E+00  7.0371604257E+02  0.0000000000E+00  -1.1081185565E-05  0.0000000000E+00  0.0000000000E+00
0.0000000000E+00  0.0000000000E+00  7.0371604257E+02  5.1208251808E-06   0.0000000000E+00  0.0000000000E+00
0.0000000000E+00  -1.1081185565E-05  5.1208251808E-06  3.5185726502E+01  0.0000000000E+00  0.0000000000E+00
1.1081185565E-05  0.0000000000E+00  0.0000000000E+00  0.0000000000E+00   1.7592863146E+01  1.1890454033E-05
-5.1208251808E-06  0.0000000000E+00  0.0000000000E+00  0.0000000000E+00  1.1890454033E-05  1.7592863356E+01
The Mass Center of the Cross Section
==================================================
Xm2 = 7.2768345058E-09
Xm3 = 1.5746671803E-08

The Mass Properties with respect to Principal Inertial Axes
==================================================
Mass Per Unit Span                      =7.0371604257E+02
Mass Moments of Intertia about x1 axis = 3.5185726502E+01
Mass Moments of Intertia about x2 axis = 1.7592851360E+01
Mass Moments of Intertia about x3 axis = 1.7592875142E+01
The user coordinate axes are the principal inertial axes.
The mass-weighted radius of gyration    =  2.2360655745E-01

The Geometric Center of the Cross Section
==================================================
Xg2 = 7.2768345083E-09
Xg3 = 1.5746671805E-08

Classical Stiffness Matrix (1-Extension; 2-Twist; 3,4-Bending)
==================================================
1.8346903296E+10  0.0000000000E+00  2.8890263597E+02  -1.3350728462E+02
0.0000000000E+00  3.4486586208E+08  0.0000000000E+00  0.0000000000E+00
2.8890263597E+02  0.0000000000E+00  4.5867185451E+08  2.8853974444E+02
-1.3350728462E+02  0.0000000000E+00 2.8853974444E+02  4.5867185713E+08


Classical Flexibility Matrix (1-Extension; 2-Twist; 3,4-Bending)
==================================================
5.4505110963E-11  0.0000000000E+00  -3.4331024793E-17  1.5865022358E-17
0.0000000000E+00  2.8996781356E-09  0.0000000000E+00  0.0000000000E+00
-3.4331024793E-17  0.0000000000E+00  2.1802078984E-09 -1.3715178393E-15
1.5865022358E-17  0.0000000000E+00  -1.3715178393E-15  2.1802078860E-09
```

Fig. B.1: The first part of one output file by VABS-GUI of example 1



```
The Neutral Axes (or Tension Center) of the Cross Section
==================================================
Xt2 = 7.2768293624E-09
Xt3 = 1.5746670231E-08

Timoshenko Stiffness Matrix (1-Extension; 2,3-Shear; 4-Twist; 5,6-Bending)
==================================================
1.8346903296E+10  0.0000000000E+00  0.0000000000E+00  0.0000000000E+00  2.8890263597E+02  -1.3350728462E+02
0.0000000000E+00  4.6819491491E+09  6.1054608679E+03  -7.5323728323E+01 0.0000000000E+00  0.0000000000E+00
0.0000000000E+00  6.1054608679E+03  4.6819577519E+09  2.8764477809E+01  0.0000000000E+00  0.0000000000E+00
0.0000000000E+00  -7.5323728323E+01  2.8764477809E+01  3.4486586208E+08 0.0000000000E+00  0.0000000000E+00
2.8890263597E+02  0.0000000000E+00  0.0000000000E+00  0.0000000000E+00  4.5867185451E+08  2.8853974444E+02
-1.3350728462E+02  0.0000000000E+00  0.0000000000E+00  0.0000000000E+00 2.8853974444E+02  4.5867185713E+08


Timoshenko Flexibility Matrix (1-Extension; 2,3-Shear; 4-Twist; 5,6-Bending)
==================================================
5.4505110963E-11  0.0000000000E+00  0.0000000000E+00  0.0000000000E+00  -3.4331024793E-17  1.5865022358E-17
0.0000000000E+00  2.1358625823E-10  -2.7852505554E-16  4.6650373578E-17  0.0000000000E+00  0.0000000000E+00
0.0000000000E+00  -2.7852505554E-16  2.1358586578E-10  -1.7814772501E-17  0.0000000000E+00  0.0000000000E+00
0.0000000000E+00  4.6650373578E-17  -1.7814772501E-17  2.8996781356E-09  0.0000000000E+00  0.0000000000E+00
-3.4331024793E-17  0.0000000000E+00  0.0000000000E+00  0.0000000000E+00  2.1802078984E-09  -1.3715178393E-15
1.5865022358E-17  0.0000000000E+00  0.0000000000E+00  0.0000000000E+00  -1.3715178393E-15  2.1802078860E-09


The Generalized Shear Center of the Cross Section in the User Coordinate System
==================================================
Xs2 = 6.1437068763E-09
Xs3 = 1.6088121300E-08
```

Fig. B.2: The second part of one output file by VABS-GUI of example 1