VABS Manual for Users*

December 20, 2021

1 Introduction

VABS (*V*ariational *A*symptotic *B*eam *S*ectional Analysis) is a code implementing the various beam theories[1–13, 13–24] based on the concept of simplifying the original nonlinear three-dimensional (3D) analysis of slender structures into a two-dimensional (2D) cross-sectional analysis and a one-dimensional (1D) nonlinear beam analysis using the powerful variational asymptotic method.[25]

VABS takes a finite element mesh of the cross-section including all the details of geometry and material as inputs to perform a homogenization analysis to compute sectional properties including inertial properties and structural properties. These properties are needed for the 1D beam analysis to predict the global behavior of the slender structure. VABS can also perform a dehomogenization analysis to compute the distributions of 3D displacements/strains/stresses, and failure indexes and strength ratios over the cross-section based on the global behavior of the 1D beam analysis.

Since most of the theoretical details are presented in pertinent papers and collected in the book by Prof. Hodges,[7] this manual will only serve to help readers get started using VABS to solve their own composite beam problems. This manual addresses the history of the code, its features, functionalities, conventions, inputs, outputs, maintenance, and tech support.

2 VABS History

The research project that gave birth to VABS was initiated by Prof. Dewey Hodges when he was first introduced to the variational asymptotic method by Prof. Berdickevsky at Georgia Tech in 1989 and has been ongoing ever since till the time of writing. The program name VABS first appeared in [1]. The original version of VABS was based on a research code written by Prof. Cesnik. Although the current version of VABS is a completely rewritten code, it is important to note that the original version played an important role for the present code and facilitated

its development. The fall semester of 1998, when Prof. Yu began his graduate study at Georgia Tech, marked the beginning of the transition of VABS from a research code to a production design and analysis tool for practicing engineers. The code was rewritten from scratch using the modern Fortran language, with all unnecessary restrictions eliminated, and the computing and memory efficiency greatly enhanced. At the same time, Prof. Cesnik was continuing his work on VABS for piezoelectric materials at MIT and later at University of Michigan. And Profs. Hodges and Yu continue their work on VABS for multiphysics modeling at Georgia Tech, Utah State University and Purdue University. For this reason, there are two variants of VABS: the Georiga Tech/Utah State/Purdue VABS, released and maintained by Profs. Yu and Hodges, and UM/VABS, released and maintained by Prof. Cesnik. From henceforth in this manual the term VABS will refer only to the Georgia Tech/Utah State/Purdue VABS, and what follows is only applicable to this code. Many researchers and engineers all over the world are actively using VABS. VABS is becoming a standard tool for design and analysis of composite slender structures such as helicopter rotor blades, wind turbine blades, high aspect ratio wings, UAM/eVTOL/UAV blades, propellers, landing gear, composite tubes, etc.

3 What is New in Different VABS versions

3.1 What is New in VABS 4.1

The new capabilities of VABS 4.1 are:

1. Perform dehomogenization for multiple load cases in terms of forces and moments corresponding to the Euler-Bernoulli model and the Timoshenko model.

3.2 What is New in VABS 4.0

The new capabilities of VABS 4.0 are:

1. Compute pointwise distributions of failure indexes and strength ratios over the cross-section under given load.

2. Compute the safety margin of the cross-section under a given load.

3. Output the nodal stress/strain values according to the original numbering of the finite element nodes.

4. Output the complete set of engineering properties commonly used in conventional beam analysis including extension stiffness (EA), torsional stiffness (GJ), principal bending stiffnesses (EI_{22}, EI_{33}), principal shear stiffnesses (GA_{22}, GA_{33}), tension center, shear center, principal inertia axis, principal bending axis, and principal shear axis.

5. Since VABS 4.0, we change the executable name to VABS.

3.3 What is New in VABS 3.9

The main feature of VABS 3.9 is prediction of sectional damping matrix based on the lamina damping coefficients. VABS 3.9 can also output the cross-sectional area. Another feature is that VABS 3.9 is now distributed as one single library.

3.4 What is New in VABS 3.8

The main feature of VABS 3.8 is a new license manager to allow more versatile license mechanisms including node locked licenses and floating licenses. The user does not have to put the license in the same folder as the input file as it was the case for previous versions. Instead, the user can use a node locked license file stored in the same folder as the VABS executable or obtain a floating license from a license server which could be the same machine the user is using or a license server on the Internet.

Since VABS 3.8, we provide free academic licenses for students and professors to use the full version of VABS for teaching and academic research.

Since VABS 3.8, we provide both Linux and Windows versions for the VABS code.

3.5 What is New in VABS 3.7

The main feature of VABS 3.7 is to carry out the recovery up to the second order which provides a better prediction for the 3D recovered fields in comparison to known exact solutions.

3.6 What is New in VABS 3.6

The main feature of VABS 3.6 is to use an improved method for optimizing the numbering of finite element mesh. For large problems, it is much faster that the method used in previous versions. The most recent version of fortran compiler is used for compiling the code resulting in much faster recovery.

3.7 What is New in VABS 3.5

The main feature of VABS 3.5 is for oblique cross-sectional analysis, the inputs are given in the oblique cross-sectional system while in previous versions, the inputs are given in the normal cross-sectional coordinates. Also starting VABS 3.5, users can use long names of input file, including spaces in the path and file names.

3.8 What is New in VABS 3.4

The main new feature is expanding $\sqrt{g}$ in the modeling for initially curved/twisted beams. For some cases such a change made significant differences for obtaining first and second correction of the stiffness matrix due to initial curvature and twist. Such a change is verified using an initially curved strip for which an elasticity solution is obtainable. The input file for this case is *isorectTrif2.sg*. A 64-bit version of VABS is also available since VABS 3.4.

3.9 What is New in VABS 3.3

The main new feature is introducing a new input format, and keeping the previous input format as optional. In the new format, the user only needs to provide one real number for θ_1 as few users take advantage of the nine real numbers for θ_1, which is useful for elements with highly curved edges. In the new format, we introduce layer definition so that a layer type instead of material type is provided for each element. Each layer is defined through a unique combination of material type and layerup angle θ_3. It is more economical than assigning θ_3 for each element, as what we have done in the previous format, because the number of layers usually is much less than the number of elements. These two changes reduce approximately 3/4 of real numbers needed for VABS inputs, saving space and time. These changes will also simplify the development of VABS preprocessors as it is easier to compute just one number for θ_1 for each element.

3.10 VABS III and What is New

VABS was originally designed to run as a stand alone code and its error handling, memory allocation/de-allocation, and I/O were handled with this use in mind. However, in recent years, more and more users began to explore the possibility of using VABS in a design environment. This motivates the major upgrade of VABS to VABS III through restructuring the code.

Since the first release of VABS III, a few users have asked the difference between VABS III and previous versions, in particular VABS 2.1.1 which was the last release and the code accompanying Prof. Hodges' book.[7] Overall, VABS III is a much improved code in both accuracy and efficiency. The main difference can be described according to the following two aspects.

- As far as functionalities concerned, VABS III

    1. Uses the correct constraints so that it can reproduce the 3D elasticity theory for isotropic prismatic beams. This change affects the warping functions, and affects all stiffness models except the classical one. Such a correction enables VABS to reproduce the 3D elasticity theory for isotropic prismatic beams and thus enables VABS to provide a better modeling for prismatic or initially curved/twisted composite beams (VABS 3.0).
    2. Recovers 3D stress/strain fields at each node in addition to the Gauss points. The recovered 3D stress/strain fields are expressed in both the beam coordinate system and the material coordinate system. VABS 2.1.1 only recovers 3D stress/strain fields at the Gauss points expressed in the beam coordinate system. For visualization, nodal values are convenient. To apply failure criteria of composite materials, stresses/strains expressed in the material coordinate system are needed (VABS 3.0).
    3. Handles isotropic, orthotropic, and anisotropic material differently. Previous versions treat all materials as orthotropic only and must take a total of 9 elastic constants. VABS III allows general anisotropic material with as many as 21 elastic constants and isotropic materials with as few as 2 elastic constants (VABS 3.0).
    4. Can model hygrothermal effects of composite beams due to temperature and moisture changes. As a companion capability, VABS Conduction is developed to carry out a di-

mensional reduction for the 3D conduction problem. VABS Conduction can be requested separately (VABS 3.1).

5. Updates the transformation procedure into the Timoshenko model from the asymptotic energy. A new perturbation method is developed to capture the effects due to initial curvatures/twist during the transformation. The prediction for Timoshenko stiffness is generally improved, even for some prismatic beams (VABS 3.2).
6. Outputs the average of 3D stresses/strains within each element for convenience of post-processing (VABS 3.2.2).
7. Provides an option for recovering the 3D displacement/strain/stress fields based on the linear beam theory (VABS 3.2.4).

- As far as the quality of the code is concerned, VABS III

1. Is restructured to change the error handling and error message handling, memory allocation and de-allocation, and I/O handling to facilitate its integration with other software environments (VABS 3.0).
2. Interprets and echoes all the input data for quicker identification of mistakes in the input file (VABS 3.0).
3. Is much faster than VABS 2.1.1 by modifying the mesh optimization algorithm and adopting a new approach to calculate the elemental finite element matrices (VABS 3.0).
4. Uses dynamic link libraries (DLLs) to encapsulate the analysis capability so that VABS has true plug-n-play capability which is convenient for integration into other environment. Now VABS can be used both as a standalone application and two callable libraries. The two callable libraries and the corresponding manual for developers can be requested separately (VABS 3.0).
5. Has more thorough and informative error handling (VABS 3.0).

Quite a few bugs in VABS 2.1.1 have been corrected in VABS III and its later versions. One bug is associated with the modified linear solver. Because of this bug, for some very rare cases, VABS 2.1.1 provides some annoying couplings which are not supposed to be there. VABS 3.0 has no such anomaly. At least one bug related with the Trapeze effect inherited from the original VABS before 1998 has been corrected in VABS 3.0. A bug related with recovery is also corrected in VABS 3.2.3.

And starting from VABS 3.0, ***an evaluation version of VABS*** is free for anybody who asks. It allows the user to evaluate the code for one month before obtaining a permanent license.

3.11 VABSII

VABS II was released in June 2004, with the major enhancement to remove the need of asking the user to choose arbitrary point constraints and let the code determine the singularity and apply the corresponding constraints. Other improvements of VABS II include calculation of principal inertial axes, the mass matrix, and neutral axes, and significant reduction of the computing time for large size problems.

4 VABS Features and Functionalities

4.1 VABS Features

Along with the features of previous versions, the most recent version of VABS has the following features:

1. It is a highly modularized code written in the modern Fortran language. All the problem-dependent arrays are allocated dynamically during run time, and the user can use all the memory up to the limit of the machine. All the outstanding abilities of array handling in the modern Fortran language have been exploited.

2. It adopts highly efficient techniques to reduce the requirement of RAM and increase the computational efficiency. Now cross-sections as complex as real composite helicopter rotor blades with hundreds of layers can be easily handled on a laptop computer.

3. It has a general element library that includes all the typical 2D elements such as 3, 4, 5, 6-noded triangular elements and 4, 5, 6, 7, 8, 9-noded quadrilateral elements. Users are free to choose the type of elements, and different types of elements can be mixed within one mesh, if necessary.

4. It can deal with arbitrary layups. Users can provide one parameter for the layup orientation and one parameter for the ply orientation to uniquely specify the material system in the global coordinate system. Nine parameters can be used for the ply orientation if a ply is highly curved and the ply angle is not uniform within an element.

5. It detects singularities and properly removes them to render the solution as a true representation of the theory. Older versions before VABS II dealt with them approximately by asking the users to input four constraints on three distinct, user-specified nodes. The arbitrariness of the older approach can affect the refined models, and sometimes may even render the linear system unsolvable.

6. It applies the four constraints on the warping functions in such a way that the 3D elasticity solution can be reproduced for isotropic beams, correcting a mistake related with these constraints in previous versions.

7. It does not require the beam reference line to be the locus of cross-sectional area centroids. VABS can calculate the centroid for any arbitrary cross-section, and users can choose their own reference line for the convenience of the 1D global beam analysis.

8. It can deal with isotropic materials, orthotropic materials, and general anisotropic materials, while all the old versions treat all materials as orthotropic.

9. It can be quickly and conveniently integrated with other environments such as computer-aided design environments, multidisciplinary optimization environments, or commercial finite element packages.

10. VABS can be executable as a stand alone executable in command line or called by other codes as a library.

4.2 VABS Functionalities

VABS is a general-purpose, cross-sectional analysis tool for computing inertial, stiffness, and strength properties of general cross-sections. Specifically, it has the following functionalities:

1. Compute the 6×6 mass matrix, written in terms of the mass per unit length, and the first and second mass moments of inertia about the three axes. Based on the information provided by the mass matrix, VABS calculates the mass center, the principal inertial axes, the principal mass moments of inertia, and the mass-weighted radius of gyration.

2. Compute the geometrical center of the cross-section and the area of the cross-section.

3. Compute the 4×4 stiffness matrix and compliance matrix for the classical model (also called the Euler-Bernoulli model) for prismatic or initially curved/twisted composite beams with normal or oblique cross-sections. Based on the classical model, VABS can calculate the location of tension center, the extension stiffness (EA), the torsional stiffness (GJ), the principal bending stiffnesses (EI_{22} and EI_{33}), and the principal bending axes.

4. Compute the 6×6 stiffness matrix and compliance matrix for the Timoshenko model for prismatic or initially curved/twisted composite beams with normal cross-sections. Based on the Timoshenko model, VABS can calculate the location of the shear center, the principal shear stiffnesses (GA_{22} and GA_{33}), and the principal shear axes.

5. Compute the 5×5 stiffness matrix and compliance matrix for the Vlasov model for prismatic or initially curved/twisted composite beams with normal cross-sections, which is important for thin-walled beams with open sections.

6. Compute the trapeze effects, a nonlinear effect important for beams under large centrifugal forces. The composite beam could be either prismatic or initially twisted and curved.

7. Compute 3D pointwise displacement, strain, and stress fields using the global behavior of a 1D global beam analysis using the classical model, the Timoshenko model, or the Vlasov model. Multiple recovery runs can be performed for different inputs of global beam responses without repeating the homogenization analysis. The recovered stress/strain fields are evaluated both at the nodal positions and Gauss points. They are expressed in both the material coordinate system and the beam coordinate system.

8. Compute sectional damping matrix for composite beams. The computation is based on the simple concept of scaling stiffness related matrices with the lamina damping coefficient specified for each material.

9. Compute hygrothermal effects of composite beams due to temperature and moisture changes. As a companion capability, VABS Conduction is developed to carry out a dimensional reduction for the 3D conduction problem, which can be requested separately.

10. Compute the failure index and strength ratio distribution over the cross-section, and the strength ration for the entire cross-section.

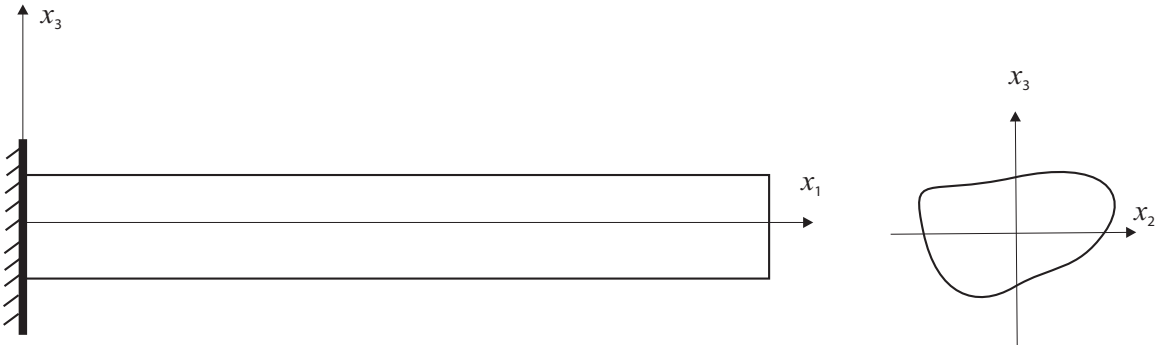

Figure 1: VABS beam coordinate system

5 VABS Conventions

To understand the inputs and interpret outputs of the program correctly, we need to explain some conventions used by VABS.

First, VABS uses a right hand system, the beam coordinate system, denoted as x_1, x_2 and x_3, with x_1 as the beam axis and x_2 and x_3 as the local Cartesian coordinates of the cross-section, see Figure 1 for a beam with an arbitrary cross-section. Usually, for rotor blades, x_1 is along the direction of the span and points to the tip, x_2 is along the direction of the trailing edge to the leading-edge of the airfoil and points to the direction of the leading edge, and x_3 points upward so that x_1, x_2, x_3 form a right-hand system. Often the origin of x_1 is located at the root of the blade, yet the user is free to choose the origin of x_2 and x_3 at an arbitrary point of the cross-section, or particular references with physical meaning such as the mass center, geometry center, tension center, or shear center. Detailed information is needed to define the cross-sectional geometric domain spanned by x_2 and x_3 and the materials that occupy that domain. Also, certain characteristics along the span direction, such as initial curvature/twist or taper, are needed for cross-sectional analyses when they are not equal to zero. The obliqueness should be specified when the angle between x_1 and the x_2-x_3 plane is not equal to 90°, that is, when reference cross-section is not normal to the reference line, such as the case of a swept wing. It is noted that the beam coordinate system is the same as the undeformed beam coordinate system b defined in Ref. [7].

Second, VABS numbers the nodes of each element in the counterclockwise direction, as shown in Figure 2 for triangular elements and Figure 3 for quadrilateral elements. Nodes 1, 2, and 3 of the triangular elements and nodes 1, 2, 3, and 4 of the quadrilateral elements are at the corners. Nodes 5, 6, 7 of the triangular elements and nodes 5, 6, 7, 8, 9 for quadrilateral elements are optional nodes.

The recovered 3D displacements are values at each node expressed in the VABS beam coordinate system (Figure 1). However, stresses and strains are most accurately evaluated at Gauss integration points. Gauss integration schemes for different orders of the two types of elements are also shown in Figures 2 and 3. The red interior points correspond to the integration scheme for linear elements and the green interior points correspond to the integration scheme for quadratic

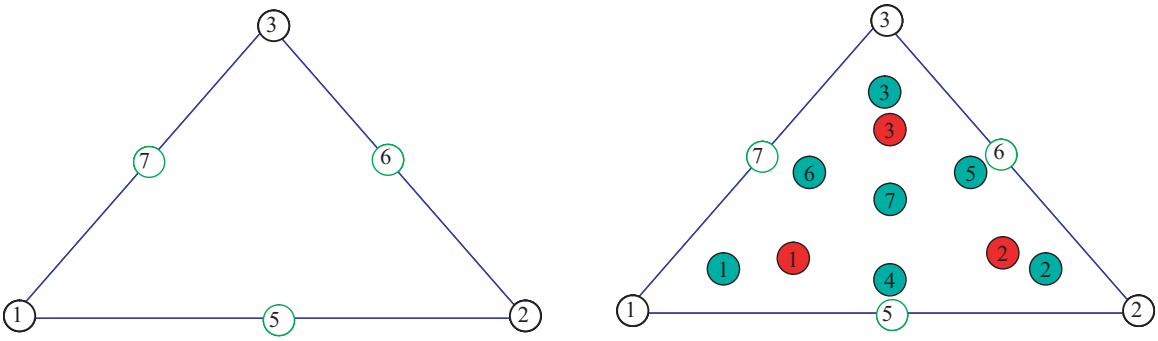

Figure 2: VABS triangular element node numbering and corresponding integration schemes

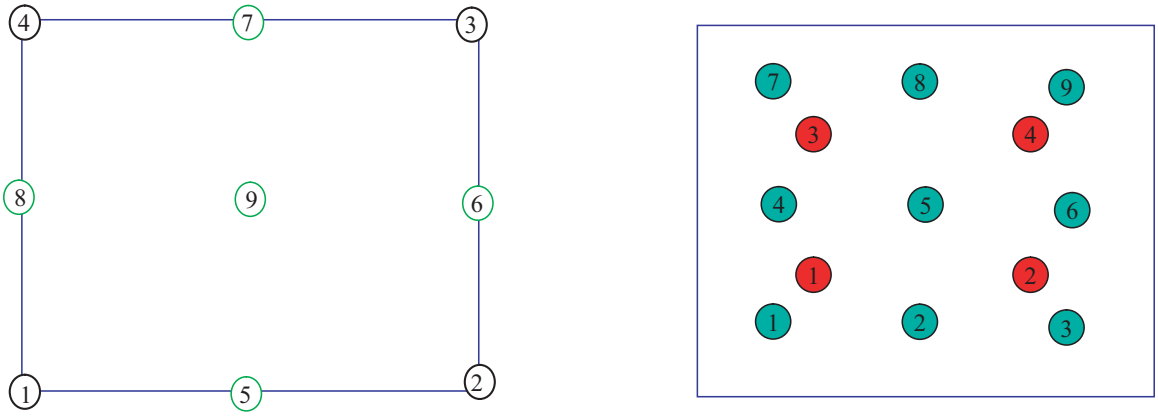

Figure 3: VABS quadrilateral element node numbering and corresponding integration schemes

elements. VABS can also recover 3D stresses and strains at each node as suggested by our industry users. The recovered stresses and strains are expressed in both the beam coordinate system and the material coordinate system which is needed for failure analysis of composite materials.

VABS allows the user to choose any unit system of convenience. However, it is necessary to be consistent in the choice of units to avoid confusion. Particularly, users must *never* use the pound as a unit of mass to avoid confusion. When pounds are used for force and feet for length, the unit of mass must be slug = lb-sec^2/ft. If inches are used for length along with pounds for force, then the unit of mass must be lb-sec^2/in.

Finally, to understand the VABS input convention for composite layups, we need to find relationships among three coordinate systems: the beam coordinate system (x_1, x_2, x_3) used by the user to define the geometry, the material system (e_1, e_2, e_3) used by the user to define the material properties, and an intermediate one to define the ply plane (y_1, y_2, y_3). As shown in Figure 4, the ply coordinate system (y_1, y_2, y_3) is formed by rotating the global coordinate system (x_1, x_2, x_3) in the right-hand sense about x_1 by the amount $0 \leq \theta_1 \leq 360°$. Then, the ply coordinate system (y_1, y_2, y_3) is rotated about y_3 in the right-hand sense by the amount $-90° \leq \theta_3 \leq 90°$ to form

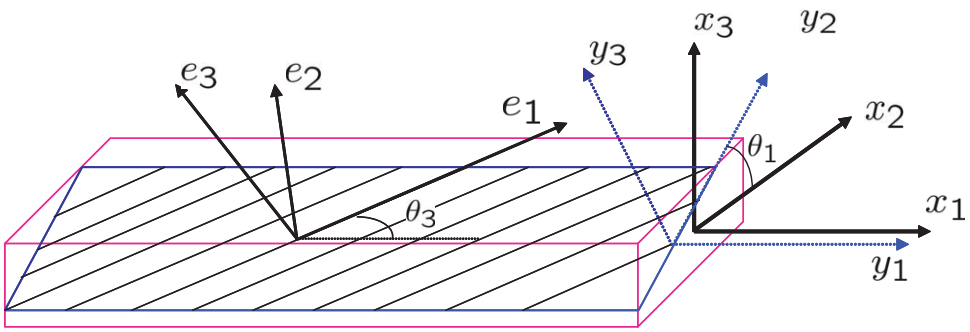

Figure 4: VABS layup convention

the material system (e_1, e_2, e_3), the range of θ_3 being the same as commonly defined in the field of composite materials. Here we use the box-beam section depicted in Figure 5 to illustrate VABS layup conventions. Here, x_1 is pointing toward the reader, x_2 is pointing to the right side of the reader, and x_3 is pointing upward vertically. For the upper wall: $\theta_1 = 0°$; the left wall: $\theta_1 = 90°$; the lower wall: $\theta_1 = 180°$; the right wall: $\theta_1 = 270°$. For all the walls $\theta_3 = \theta$ for the box-beam in Figure 5 because all the fibers are rotating positively about y_3/e_3 by the angle θ. The users can specify their own stacking sequences. The stacking sequences expressed from the innermost layer to the outermost layer for each wall are often used.

6 VABS Installation and Execution

VABS is distributed in the form of VABSx.xReleasePCMM-DD-YEAR.exe for Windows operating systems with "x.x" denotes the version number. Double click the file VABSx.xReleasePCMM-DD-YEAR.exe and follow the setup wizard is all you need to do for installation. More details can refer to the Readme file. You should add the folder where you stored VABS executable into the Path.

VABS is a command line code. It can be executed using *VABS inputfile analysis nload* in the regular command prompt. The command line argument *analysis* lets VABS know which type of analyses to perform.

1. If *analysis* does not exist, VABS will carry out a homogenization analysis to compute the inertial and stiffness properties.

2. If *analysis* is equal to 1, VABS will carry out a dehomogenization analysis to recover 3D displacements, strains, stresses based on the nonlinear beam theory.

3. If *analysis* is equal to 2, VABS will carry out a dehomogenization analysis to recover 3D displacements, strains, stresses based on the linear beam theory.

4. If *analysis* is equal to 3, VABS will carry out a dehomogenization analysis to evaluate the distribution of failure index and strength ratio over the cross-section, and the strength ratio of the entire cross-section.

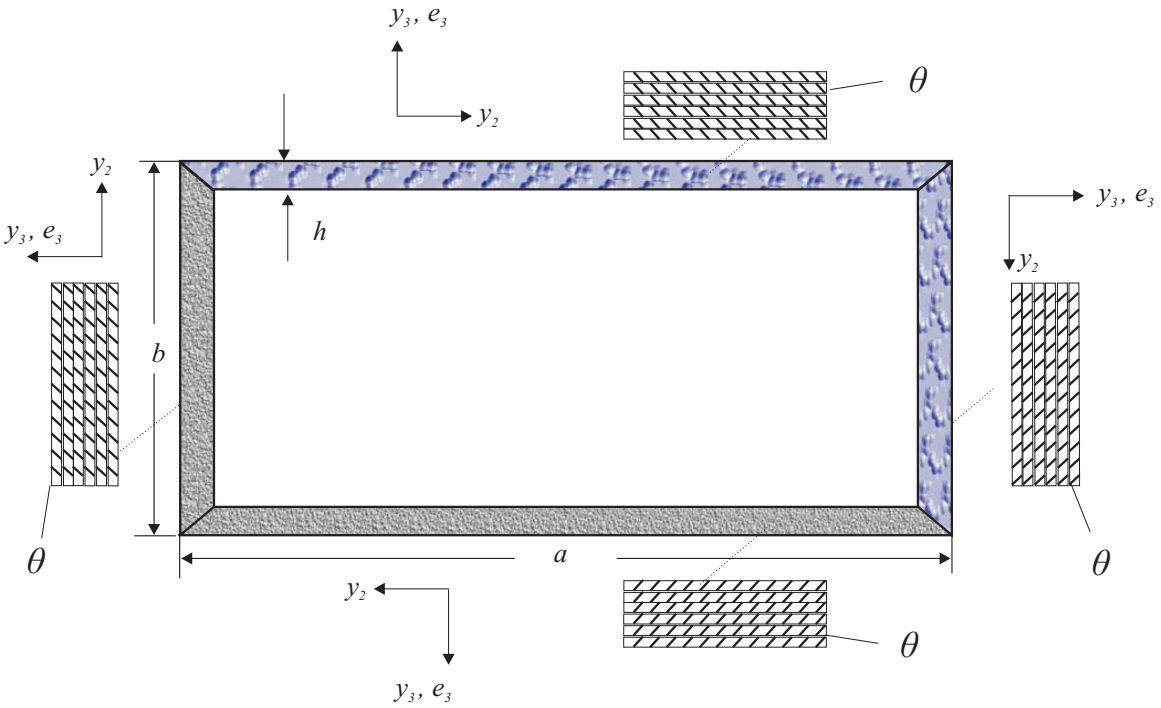

Figure 5: VABS layup convention for a box-beam

If *analysis* is equal to 1, 2, 3, we can also provide another command line argument *nload* so that VABS knows to perform the corresponding dehomogenization analysis for how many load cases. If this argument does not exist, it will perform the dehomogenization analysis for a single load case.

This command line execution can be achieved by Click Start, choose Run, and type in "cmd" and click OK. Then use "cd" to enter the right folder where the input file is in, then type *VABS inputfile argu*. If the input file number includes spaces, one should use *VABS "inputfile" analysis nload*. If the input file is not located in the current folder, you should use *VABS "absolute path\inputfile" analysis nload* to let VABS correctly locate the input file. More details for running the code on a Windows machine can be found in the Readme file.

The Linux version of VABS is distributed in the form of VABSx.xReleaseLinuxMM-DD-YEAR.zip with "x.x" denotes the version number. Unzip the file VABSx.xReleaseLinuxMM-DD-YEAR.zip into a folder of your choice is all you need to do for installing VABS. The command to execute VABS on a Linux machine is the same as that on a Windows machine. More details can be found in the Readme file.

7 VABS Inputs

Although a few preprocessors, such as PreVABS, have been developed to create VABS input files, it is still beneficial for advanced users, particularly those who want to embedded VABS in their own design environment, to understand the meaning of the input data.

Starting from VABS 4.0, the inputs for the VABS are separated into two files: homogenization input file and dehomogenization input file. VABS homogenization run only requires the homogenization input file with a name of the user's choice. The dehomogenization input file associated with the homogenization input file with extension *glb*. In other words, if the homogenization input file name is *input_file_name*, the dehomogenization input file must be *input_file_name.glb*.

7.1 Homogenization Input File

The first line lists two newly introduced integer flags arranged as: "*format_flag nlayer*". If the first flag is 1, the input is prepared in the new format, otherwise, it is prepared in the old format. The second integer provides the number of layers in the section. Note, here layer is defined as a unique combination of material type and layup orientation, it does not necessarily corresponds to the definition used in the manufacturing sense. For example, even if a section is made of a single isotropic material, we consider it has one layer. Hence, *nlayer* should be always given a value greater than one if *format_flag*=1. Note *nlayer* is not used in the old format.

The second line has two flags arranged as: "*Timoshenko_flag damping_flag thermal_flag*". The first flag, *Timoshenko_flag*, can be only 1 or 0. If it is 1, VABS will construct both the classical model (also called the Euler-Bernoulli model) and the Timoshenko model. If it is 0, it will only construct the classical model. The second flag, *damping_flag*, can be equal to 0 or 1. If it is equal to 0, VABS will not compute the damping matrix for the section. If it is equal to 1, VABS will compute the damping matrix. The third flag, *thermal_flag*, can be equal to 0 or 3. If it is equal to zero, VABS will carry out a pure mechanical analysis. If it is equal to 3, VABS will carry out a one-way coupled thermoelastic analysis.

The third line has four flags arranged as: "*curve_flag oblique_flag trapeze_flag Vlasov_flag*." These flags can be only 1 or 0. Their uses are explained in the following:

1. To model initially curved and twisted beams, *curve_flag* is 1, and three real numbers for the twist (k_1) and curvatures (k_2 and k_3) should be provided in the very next line.

2. To model oblique cross-sections, *oblique_flag* is *1*, and two real numbers are needed in the following line to specify the orientation of an oblique reference cross-section, see Figure 6 for a sketch of such a cross-section. The first number is cosine of the angle between normal of the oblique section (y_1) and beam axis x_1. The second number is cosine of the angle between y_2 of the oblique section and beam axis (x_1). The summation of the square of these two numbers should not be greater than 1.0 in double precision. The inputs including coordinates, material properties, etc. and the outputs including mass matrix, stiffness matrix, etc. are given in the oblique system, the y_i coordinate system as shown in Figure 6. Note that this feature is only enabled for the classical beam model.

3. To obtain the trapeze effect, *trapeze_flag* is 1.

4. To obtain the Vlasov model, *Vlasov_flag* is 1. *Vlasov_flag* can be 1 only if *Timoshenko_flag* is 1. VABS will first construct the Timoshenko model, which determines the location of the shear center. If the shear center is not at the origin of the beam coordinate system, VABS

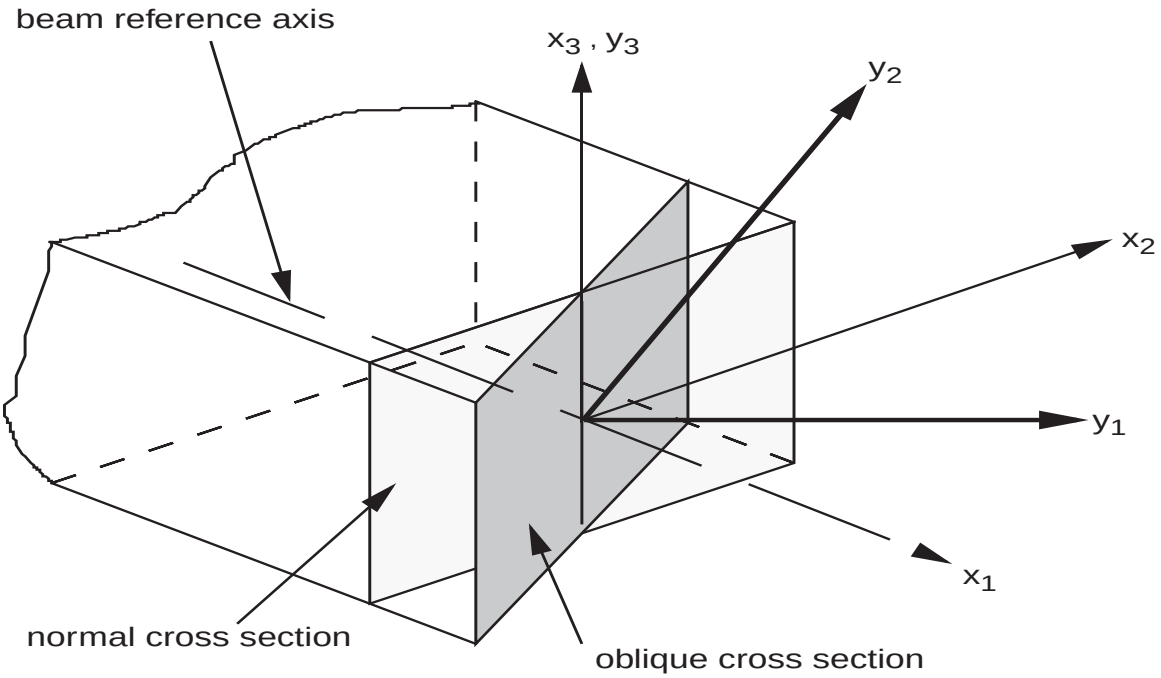

Figure 6: Sketch of an oblique reference cross-section

will move the origin of beam coordinate system to the shear center and repeat the calculation to obtain the Vlasov model.

The next line lists three integers arranged as: "*nnode nelem nmate*," where *nnode* is the total number of nodes, *nelem* the total number of elements, and *nmate* the total number of material types.

The next *nnode* lines are the coordinates for each node arranged as: "*node_no* x_2 x_3," where *node_no* is an integer representing the unique number assigned to each node and x_2, x_3 are two real numbers describing the location (x_2, x_3) of the node. **Although the arrangement of *node_no* is not necessary to be consecutive, every node starting from 1 to *nnode* should be present.**

The next *nelem* lines list 10 integers for the nodes for each element (also known as the connectivity relations). They are arranged as: "*elem_no node_1 node_2 node_3 node_4 node_5 node_6 node_7 node_8 node_9*," where *elem_no* is the number of element and *node_i* ($i = 1, 2, \ldots, 9$) are nodes of this element. If a node is not present in the element, the value is 0. If *node_4* is 0, the element is a triangular element; see Figures 2 and 3 for the VABS numbering convention. **Although the arrangement of *elem_no* is not necessary to be consecutive, every element starting from 1 to *nelem* should be present.**

If $format\_I = 1$, that is, if the new format is used, the next *nelem* lines list the layer type and the layer plane angle (θ_1) for each element as: *elem_no layer_type* θ_1, where *layer_type* is an integer representing which layer the element *elem_no* belongs to, and θ_1 is a real number describing the layer plane angle for the element. Here, θ_1 is assumed to be constant for each element, thus it can

be calculated at any material point belonging to the element, such as the centroid, or computed as the average of θ_1 of all the points within the element. **Although the arrangement of *elem_no* is not necessary to be consecutive, every element starting from 1 to *nelem* should be present.** For isotropic materials, θ_1 will not enter the calculations.

If $format\_I$ is not equal to 1, that is, if the old format is used, the next *nelem* lines list the material type and layup parameters for each element as: *elem_no material_type* θ_3 $\theta_1(9)$, where *material_type* is an integer representing the type of the material for the element *elem_no*, θ_3 is a real number representing the layup angle in degrees for this element, and $\theta_1(9)$ is an array storing nine real numbers for the layer plane angles at the nodes of this element. For simplification, if the ply orientation can be considered as uniform for this element, $\theta_1(1)$ stores the layer plane angle and $\theta_1(2) = 540°$, and all the rest can be zeros or other real numbers because they do not enter the calculation. If the element has fewer than nine nodes, zeros are to be input for the corresponding missing nodes, as in the case for connectivity. **Although the arrangement of *elem_no* is not necessary to be consecutive, every element starting from 1 to *nelem* should be present.** For isotropic materials, neither θ_3 nor $\theta_1(9)$ will enter the calculations.

If $format\_I = 1$, that is, if the new format is used, the next *nlayer* lines define the layers used in the section. They are arranged as: *layer_id mate_type* θ_3, where *layer_id* is an integer denoting the identification number for each layer, *mate_type* is an integer denoting the material type used in the layer, and θ_3 is a real number denoting the layup orientation. For example, if layer 1 is made of material 1 and having $-15°$ layup, we will provide the information as $1 \quad 1 \quad -15.0$. If *damping_flag* is 1, a damping coefficient for each layer is also needed to input right after θ_3. In others words, the input should be arranged as *layer_id mate_type* θ_3, *damping_layer*, with *damping_layer* indicating the damping coefficient for the layer.

The next *nmate* blocks defines the material properties. They are arranged as:
mat_id orth
const1 const2 ....
where *mat_id* is the number of material type, *orth* is the flag to indicate whether the material is isotropic (0), orthotropic (1) or general anisotropic (2). The rest are material constants.

For isotropic materials, *orth* is 0, if *thermal_flag* is 0, there are 3 constants arranged as:
$E \quad \nu$
ρ
where E is the Young's modulus, ν is the Poisson's ratio, and ρ is the density of the material. Poisson's ratio must be greater than -1.0 and less than 0.5 for linearly elastic isotropic materials, although VABS allows users to input values that are very close to those limits.

If *thermal_flag* is 3 and *orth* is 0, and there are 4 constants arranged as:
$E \quad \nu$
ρ
α
where α is the coefficient of thermal expansion (CTE).

For orthotropic materials, *orth* is 1, if *thermal_flag* is 0, there are 10 constants arranged as:

E_1 E_2 E_3

G_{12} G_{13} G_{23}

ν_{12} ν_{13} ν_{23}

ρ

including the Young's moduli (E_1, E_2, and E_3), the shear moduli (G_{12}, G_{13}, and G_{23}), the Poisson's ratios (ν_{12}, ν_{13}, and ν_{23}), and the mass density (ρ). The convention of values is such that these values will be used to form the following the Hooke's law for composite materials:

$$\begin{Bmatrix} \varepsilon_{11} \\ 2\varepsilon_{12} \\ 2\varepsilon_{13} \\ \varepsilon_{22} \\ 2\varepsilon_{23} \\ \varepsilon_{33} \end{Bmatrix} = \begin{bmatrix} \frac{1}{E_1} & 0 & 0 & -\frac{\nu_{12}}{E_1} & 0 & -\frac{\nu_{13}}{E_1} \\ 0 & \frac{1}{G_{12}} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{G_{13}} & 0 & 0 & 0 \\ -\frac{\nu_{12}}{E_1} & 0 & 0 & \frac{1}{E_2} & 0 & -\frac{\nu_{23}}{E_2} \\ 0 & 0 & 0 & 0 & \frac{1}{G_{23}} & 0 \\ -\frac{\nu_{13}}{E_1} & 0 & 0 & -\frac{\nu_{23}}{E_2} & 0 & \frac{1}{E_3} \end{bmatrix} \begin{Bmatrix} \sigma_{11} \\ \sigma_{12} \\ \sigma_{13} \\ \sigma_{22} \\ \sigma_{23} \\ \sigma_{33} \end{Bmatrix}$$

If *thermal_flag* is 3 and *orth* is 1, and there are 13 constants arranged as:

E_1 E_2 E_3

G_{12} G_{13} G_{23}

ν_{12} ν_{13} ν_{23}

ρ

α_{11} α_{22} α_{33}

where $\alpha_{11}, \alpha_{22}, \alpha_{33}$ are the CTEs along three directions.

The material constants are expressed in the material coordinate system; see Figure 4. It is also emphasized that if the users are provided material properties in a different coordinate system, or the arrangement of stresses and strains are different from what VABS uses, a proper transformation of the material properties is needed.

For general anisotropic materials, *orth* is 2, if *thermal_flag* is 0, there are 22 constants arranged as:

$$\begin{array}{cccccc} c_{11} & c_{12} & c_{13} & c_{14} & c_{15} & c_{16} \\ & c_{22} & c_{23} & c_{24} & c_{25} & c_{26} \\ & & c_{33} & c_{34} & c_{35} & c_{36} \\ & & & c_{44} & c_{45} & c_{46} \\ & & & & c_{55} & c_{56} \\ & & & & & c_{66} \\ & & & & & \rho \end{array}$$

These values are defined using the following Hooke's law:

$$\begin{Bmatrix} \sigma_{11} \\ \sigma_{12} \\ \sigma_{13} \\ \sigma_{22} \\ \sigma_{23} \\ \sigma_{33} \end{Bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} & c_{15} & c_{16} \\ c_{12} & c_{22} & c_{23} & c_{24} & c_{25} & c_{26} \\ c_{13} & c_{23} & c_{33} & c_{34} & c_{35} & c_{36} \\ c_{14} & c_{24} & c_{34} & c_{44} & c_{45} & c_{46} \\ c_{15} & c_{25} & c_{35} & c_{45} & c_{55} & c_{56} \\ c_{16} & c_{26} & c_{36} & c_{46} & c_{56} & c_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_{11} \\ 2\varepsilon_{12} \\ 2\varepsilon_{13} \\ \varepsilon_{22} \\ 2\varepsilon_{23} \\ \varepsilon_{33} \end{Bmatrix}$$

If *thermal_flag* is 3 and *orth* is 2, there are 28 constants arranged as:

c_{11} c_{12} c_{13} c_{14} c_{15} c_{16}
c_{22} c_{23} c_{24} c_{25} c_{26}
c_{33} c_{34} c_{35} c_{36}
c_{44} c_{45} c_{46}
c_{55} c_{56}
c_{66}
ρ
α_{11} $2\alpha_{12}$ $2\alpha_{13}$ α_{22} $2\alpha_{23}$ α_{33}

where α_{ij}, with $i = 1, 2, 3$ and $j = 1, 2, 3$, are the components of the second-order CTE tensor. CTEs corresponding to the shear strains are multiplied by two because the engineering shear strains are twice of the corresponding tensorial shear strains. Again, the material constants are expressed in the material coordinate system; see Figure 4. It is also emphasized that if the users are provided material properties in a different coordinate system, or the arrangement of stresses and strains are different from what VABS uses, a proper transformation of the material properties is needed.

If *damping_flag* is 1, a damping coefficient is input on the very next line following the density input. For example, if *orth*=0 and *thermal_flag*=3 (thermoelastic analysis with isotropic materials), the material constants are arranged as:

E ν
ρ
γ
α

where γ is a scalar representing the material damping property. It is noted that the damping coefficients for each layer and for each material are additive. In other words, the total damping coefficient used to scale the stiffness-related matrices is $damping\_layer + \gamma$.

If *thermal_flag* is equal to 3, we also need to provide the following *nnode* lines for temperature for each node arranged as: "*node_no* T," where *node_no* is an integer representing the unique number assigned to each node and T is a real number describing the temperature of the node. These temperature values can be calculated either from a 3D heat conduction analysis or using VABS Conduction, which is a generalization of the VABS approach for heat conduction analysis. VABS Conduction can be requested separately. **Although the arrangement of *node_no* is not necessary to be consecutive, every node starting from 1 to *nnode* should be present.**

Now, we have prepared all the inputs necessary for performing the homogenization run to compute the inertial properties and structural properties of the cross-section. That is, when *argu*

in Section 6 does not exist.

7.2 Dehomogenization Input File

If *argu* in Section 6 is equal to 1,2, 3, users should provide additional information in the dehomogenization input file for VABS to perform a dehomogenization analysis. A corresponding homogenization analysis must be run before carrying out the dehomogenization analysis.

If *argu* is equal to 3, VABS will perform failure analysis of the cross-section. Strength properties for each material must be provided in the dehomogenization input file. Strength properties for each material include a failure criterion and corresponding strength constants. Two lines will be inserted and the inputs needed for failure analyses should be arranged as:
$failure\_criterion \;\; num\_of\_constants$
$const_1 \;\; const_2 \;\; const_3 \;\; \ldots$
$failure\_criterion$ is an integer identifier for the failure criterion. $num\_of\_constants$ indicates the number of strength constants needed for the corresponding failure criterion. $const_1 \;\; const_2 \;\; const_3 \ldots$ are the corresponding strength constants. It is noted that this block of data should be corresponding to the material block in the homogenization input file. In other words, for each material with *mat_id*, we need to provide such information.

$failure\_criterion$ can be equal to 1, 2, 3, 4, 5, and another number greater than 10. For isotropic material, 1 is max principal stress criterion, 2 is max principal strain criterion, 3 is max shear stress criterion (also commonly called Tresca criterion), 4 max shear strain criterion, and 5 is Mises criterion. For anisotropic materials, 1 is max stress criterion for anisotropic materials, 2 is max strain criterion for anisotropic materials, 3 is Tsai-Hill criterion, 4 is Tsai-Wu criterion and 5 is Hashin criterion. 11 and larger indicates a user-defined failure criterion. It is assumed that the number of strength constants will not be greater than 9 for a material. If the material is isotropic, the failure criterion and corresponding strength constants are defined as follows:

- If $failure\_criterion$ is 1, the max principal stress criterion is used and two strength constants are needed: one for tensile strength (X) and one for compressive strength (X'), arranged as X, X'.
- If $failure\_criterion$ is 2, the max principal strain criterion is used and two strength constants are needed: one for tensile strength (X_ε) and one for compressive strength (X'_ε), arranged as $X_\varepsilon, X'_\varepsilon$.
- If $failure\_criterion$ is 3, the max shear stress criterion (aka Tresca criterion) is used and one shear strength constant (S) is needed.
- If $failure\_criterion$ is 4, the max shear strain criterion is used and one shear strength constant (S_ε) is needed.
- If $failure\_criterion$ is 5, the Mises criterion is used and one strength constant (X) is needed.

If the material is **not isotropic** (transversely isotropic, orthotropic, or general anisotropic), the failure criterion and corresponding strength constants are defined as follows:

- If $failure\_criterion$ is 1, the max stress criterion is used and nine strength constants are needed: three tensile strengths (X, Y, Z) in three directions, three compressive strengths (X', Y', Z') in three directions, and three shear strengths (R, T, S) in three principal planes, arranged as $X, Y, Z, X', Y', Z', R, T, S$.
- If $failure\_criterion$ is 2, the max strain criterion is used and nine strength constants are needed: three tensile strengths $(X_\varepsilon, Y_\varepsilon, Z_\varepsilon)$ in three directions, three compressive strengths $(X'_\varepsilon, Y'_\varepsilon, Z'_\varepsilon)$ in three directions, and three shear strengths $(R_\varepsilon, T_\varepsilon, S_\varepsilon)$ in three principal planes, arranged as $X_\varepsilon, Y_\varepsilon, Z_\varepsilon, X'_\varepsilon, Y'_\varepsilon, Z'_\varepsilon, R_\varepsilon, T_\varepsilon, S_\varepsilon$.
- If $failure\_criterion$ is 3, the Tsai-Hill criterion is used and six strength constants are needed: three normal strengths in three directions and three shear strengths in three principal planes, arranged as X, Y, Z, R, T, S.
- If $failure\_criterion$ is 4, the Tsai-Wu criterion is used and nine strength constants are needed: three tensile strengths (X, Y, Z), three compressive strengths (X', Y', Z') in three directions, and three shear strengths (R, T, S) in three principal planes, arranged as $X, Y, Z, X', Y', Z', R, T, S$.
- If $failure\_criterion$ is 5, the Hashin criterion is used and six strength constants are needed: two tensile strengths (X, Y), two compressive strengths (X', Y') in two directions, and two shear strengths (R, S) in two principal planes, arranged as X, Y, X', Y', R, S.

It is noted that for failure analyses, general anisotropic materials are also approximated using orthotropic materials due to limited number of strength constants. In VABS, both the tensile strengths and compressive strengths are expressed using positive numbers. In other words, in the uniaxial compressive test along y_1 direction, $\sigma_{11} = -X'$ when material fails.

The above block of data for strength properties for each material only needed if *argu*=3. If *argu*=1 or 2, the strength properties are not needed and should not be provided in the dehomogenization input file. Only the global beam responses as explained below should be stored in the dehomogenization input file.

The rest of inputs in the dehomogenization input file contains the global beam responses obtained from the 1D global beam analysis. To carry out a dehomogenization analysis based on the classical model, VABS requires the following data:

$$\begin{array}{cccc} u_1 & u_2 & u_3 & \\ C_{11} & C_{12} & C_{13} & \\ C_{21} & C_{22} & C_{23} & \\ C_{31} & C_{32} & C_{33} & \\ F_1 & M_1 & M_2 & M_3 \end{array}$$

where u_1, u_2, and u_3 are the 1D beam displacements along x_1, x_2, x_3, respectively. The matrix C_{ij}, with $i = 1, 2, 3$ and $j = 1, 2, 3$, is the direction cosine matrix defined as

$$\mathbf{B}_i = C_{i1}\mathbf{b}_1 + C_{i2}\mathbf{b}_2 + C_{i3}\mathbf{b}_3 \quad \text{with} \quad i = 1, 2, 3$$

where $\mathbf{B}_1$, $\mathbf{B}_2$, and $\mathbf{B}_3$ are the base vectors of the deformed beam and $\mathbf{b}_1$, $\mathbf{b}_2$, and $\mathbf{b}_3$ are the base vectors of the undeformed beam. Details of this definition can be found in Ref. [7]. u_i and C_{ij} are

needed only for recovering 3D displacements. If the user is not interested in 3D displacements, these values can be arbitrary real numbers. F_1 is the axial force, M_1 is the torque, M_2 is the bending moment around x_2, and M_3 is the bending moment around x_3. The sectional stress resultants are needed for computing 3D stresses/strains/failure indexes/strength ratios within the cross-section. For example, if the user wants to compute these quantities under 1 unit tensile axial force along with 1 unit bending moment around x_2, the inputs can be arranged as:

$$\begin{array}{cccc} 0 & 0 & 0 & \\ 1 & 0 & 0 & \\ 0 & 1 & 0 & \\ 0 & 0 & 1 & \\ 1 & 0 & 1 & 0 \end{array}$$

To perform dehomogenization for multiple load cases, the user needs to insert corresponding lines of F_1, M_1, M_2, M_3 after the end of this block. For example, to perform dehomogenization for two more load cases with $F_1 = 2, M_1 = 2, M_2 = M_3 = 0$ and $F_1 = 2, M_1 = 3, M_3 = 4, M_5 = 5$, we must provide the following inputs.

$$\begin{array}{cccc} 0 & 0 & 0 & \\ 1 & 0 & 0 & \\ 0 & 1 & 0 & \\ 0 & 0 & 1 & \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 2 & 3 & 4 & 5 \end{array}$$

To carry out a dehomogenization analysis based on the Timoshenko model, VABS requires the following data:

$$\begin{array}{cccccc} u_1 & u_2 & u_3 & & & \\ C_{11} & C_{12} & C_{13} & & & \\ C_{21} & C_{22} & C_{23} & & & \\ C_{31} & C_{32} & C_{33} & & & \\ F_1 & M_1 & M_2 & M_3 & & \\ F_2 & F_3 & & & & \\ f_1 & f_2 & f_3 & m_1 & m_2 & m_3 \\ f_1' & f_2' & f_3' & m_1' & m_2' & m_3' \\ f_1'' & f_2'' & f_3'' & m_1'' & m_2'' & m_3'' \\ f_1''' & f_2''' & f_3''' & m_1''' & m_2''' & m_3''' \end{array}$$

where the additional data F_2 and F_3 are transverse shear forces along x_2 and x_3, respectively. f_1, f_2, f_3 are distributed forces (including both applied forces and inertial forces) per unit span along x_1, x_2, x_3 respectively. m_1, m_2, m_3 are distributed moments (including both applied and inertial moments) per unit span along x_1, x_2, x_3 respectively. The prime denotes derivative with respect to beam axis, that is $()' = \frac{\partial}{\partial x_1}$, $()'' = \frac{\partial^2}{x_1^2}$, and $()''' = \frac{\partial^3}{x_1^3}$. If $nload > 1$, at the end of the above data block, we need to append two lines (one line for F_1, M_1, M_2, M_3 and one line for F_2, F_3) for each load case.

To carry out a dehomogenization analysis based on the Vlasov model, VABS requires the following data:

$$\begin{array}{ccccccc} u_1 & u_2 & u_3 & & & & \\ C_{11} & C_{12} & C_{13} & & & & \\ C_{21} & C_{22} & C_{23} & & & & \\ C_{31} & C_{32} & C_{33} & & & & \\ \bar{\gamma}_{11} & \bar{\kappa}_1 & \bar{\kappa}_2 & \bar{\kappa}_3 & \bar{\kappa}_1' & \bar{\kappa}_1'' & \bar{\kappa}_1''' \end{array}$$

where $\bar{\gamma}_{11}$ is the beam axial strain, $\bar{\kappa}_1$ is the twist , $\bar{\kappa}_2$ and $\bar{\kappa}_3$ are the curvatures around x_2 and x_3 respectively. *It is noted that the global behavior needed for dehomogenization analyses should not violate the small strain assumption. Otherwise, you might get some unexpected results. For example, if your transverse shear stiffness is 2.5 N, then inputting a shear force resultant of 1 N is too large as the shear strain will be about 0.4, which cannot be considered as small, the basic assumption of the VABS theory.*

Both input files, *input_file_name* and *input_file_name.glb*, should be ended with a blank line to avoid any possible incompatibility of different computer operating systems. The input file can be given any name as long as the total number of the characters of the name including extension is not more than 256. For the convenience of the user to identify mistakes in the input file, all the inputs are echoed in the file named *input_file_name.ech*. Error messages are also written at the end of *input_file_name.ech* and on the output screen.

8 VABS Outputs

VABS homogenization analysis outputs the sectional properties stored in a text file named *input_file_name.K*. VABS dehomogenization analysis could output 3D displacement/strain/stress, or failure index/strength ratio distributions over the cross-section in different files as explained later. All these output files are in pure text format and can be opened by any text editor.

8.1 Homogenization Outputs

Sectional properties obtained by a VABS homogenization analysis are stored in *input_file_name.K*. Some results are listed as individual numbers, and some are listed as matrices. The definitions of these properties are briefly summarized here for the convenience of end users. For more details, please refer to VABS related publications.

VABS first computes the inertial properties which is represented by a 6×6 mass matrix with respect to the beam coordinate system. The elements of the mass matrix are arranged as

$$\begin{bmatrix} \mu & 0 & 0 & 0 & \mu x_{M3} & -\mu x_{M2} \\ 0 & \mu & 0 & -\mu x_{M3} & 0 & 0 \\ 0 & 0 & \mu & \mu x_{M2} & 0 & 0 \\ 0 & -\mu x_{M3} & \mu x_{M2} & i_{22}+i_{33} & 0 & 0 \\ \mu x_{M3} & 0 & 0 & 0 & i_{22} & i_{23} \\ -\mu x_{M2} & 0 & 0 & 0 & i_{23} & i_{33} \end{bmatrix}$$

where μ is mass per unit length, x_{M2} and x_{M3} are the two coordinates of the mass center (also called the mass-weighted centroid), and i_{22}, i_{23} and i_{33} are the second mass moments of inertia. The mass center and mass moments of inertia are measured with respect to the origin O and coordinate axes x_2 and x_3. VABS also outputs the mass center, the mass matrix measured with respect to the coordinate system with the mass center as the origin and coordinates parallel to the beam coordinate system. Furthermore, VABS also outputs the inertial properties with respect to the principal inertial coordinate system (origin at the mass center, coordinates aligning with the principal inertial axes) including a mass per unit length, mass moments of inertia about the three axes, the orientation of the principal inertial axes, and mass-weighted radius of gyration (defined as the square root of the mass moment of inertia about x_1 divided by the mass per unit length).

VABS outputs the geometric center and the area of the cross-section. Only the geometry occupied by a material enters the calculation. In other words, if the cross-section is made of a single material, the geometric center is located at the same location as the mass center and the tension center.

VABS outputs the 4×4 stiffness matrix and compliance matrix for the classical beam model with respect to the beam coordinate system. Based on the compliance matrix, VABS also outputs the tension center (also called the modulus-weighted centroid), extension stiffness (commonly denoted as EA), torsional stiffness (commonly denoted as GJ), principal bending stiffnesses (commonly denoted as EI_{22}, E_{33}), and the orientation of the principal bending axes. The principal bending stiffnesses are computed with respect to the tension center and the principal bending axes.

If *damping_flag* is equal to 1, VABS also outputs the 4×4 damping matrix for the cross-section.

If *thermal_flag* is equal to 3, VABS also output the thermal stress resultants and thermal strains due to temperature changes for the classical beam model.

If *curve_flag* is equal to 1, VABS also output the stiffness matrix, compliance matrix, tension center, tension center, extension stiffness, torsional stiffness, principal bending stiffnesses, and the orientation of the principal bending axes modified by the initial curvatures/twist of the beam.

If *Timoshenko_flag* is equal to 1, VABS outputs the 6×6 stiffness matrix and compliance matrix for the Timoshenko beam model with respect to the beam coordinate system. Based on the compliance matrix, VABS also outputs the shear center (also called the twist center), principal shear stiffnesses, and the orientation of the principal shear axes. The principal shear stiffnesses are computed with respect to the shear center and the principal shear axes. If *damping_flag* is equal to 1, VABS also outputs the 6×6 damping matrix for the cross-section.

If *Vlasov_flag* is equal to 1, VABS outputs the 5×5 stiffness matrix and compliance matrix for the Vlasov beam model with respect to the beam coordinate system. This model is important for capturing the "restrained warping" effect for thin-walled beams with open sections. For such beams, it is meaningful to have a Vlasov model based on choosing the shear center as its reference. Hence to obtain a Vlasov model, VABS finds the shear center first and then shifts the origin of the

coordinate system to the shear center and calculate the 5×5 stiffness matrix and compliance matrix. If *damping_flag* is equal to 1, VABS also outputs the 5×5 damping matrix for the cross-section.

If *trapeze_flag* is equal to 1, VABS output four 4×4 coefficient matrices associated with the four classical deformation modes (extension, twist, and two bending modes) for capturing the trapeze effect important for torsionally soft rotating beams.

8.2 Dehomogenization Outputs

If *argu* is equal to 1 or 2, VABS will carry out a dehomogenization analysis to recover 3D displacements/strains/stresses based on the nonlinear or linear beam theory. The recovered 3D displacements are stored in *input_file_name.U*. The values are listed for each node identified by its location as: "x_2 x_3 u_1 u_2 u_3", where x_2 and x_3 are the coordinates of the node, u_i the recovered 3D displacements at this node, expressed in the beam coordinate system.

The recovered 3D strains for each Gauss point measured in the beam coordinate system are stored in *input_file_name.E*. The values are listed for each Gauss point identified by its location as: "x_2 x_3 ε_{11} $2\varepsilon_{12}$ $2\varepsilon_{13}$ ε_{22} $2\varepsilon_{23}$ ε_{33}", where ε_{ij} are the 3D strain components. The recovered 3D strains measured in the material coordinate system are stored in *input_file_name.EM*.

The recovered 3D stresses for each Gauss point measured in the beam coordinate system are stored in *input_file_name.S*. The values are listed for each Gauss point identified by its location as: "x_2 x_3 σ_{11} σ_{12} σ_{13} σ_{22} σ_{23} σ_{33}", where σ_{ij} are the 3D stress components. The recovered 3D stresses measured in the material coordinate system are stored in *input_file_name.SM*.

The recovered 3D strains for each node measured in the beam coordinate systems are stored in *input_file_name.EN*. The values are listed for each node identified by its node number and the location as: " *node_no* x_2 x_3 ε_{11} $2\varepsilon_{12}$ $2\varepsilon_{13}$ ε_{22} $2\varepsilon_{23}$ ε_{33}", where *node_no* denotes the node number originally given in the homogenization input file. The node number is arranged consecutively from 1 to the total number of nodes. Multiple strain values could exist for one node if the node is shared by multiple elements. The recovered 3D strains for each node measured in the material coordinate system are stored in *input_file_name.EMN*. The corresponding recovered 3D stresses for each node measured in the beam coordinate system and the material coordinate system are stored in *input_file_name.SN* and *input_file_name.SMN*, respectively.

The average of 3D strains and stresses among all the Gauss points within each element are stored in *input_file_name.ELE*, where the integer number indicating the element number, the following six real numbers are strains measured in the beam coordinate system, the next six real numbers are the stresses measured in the beam coordinate system, the next six real numbers are strains measured in the material coordinate system, the last six real numbers are the stresses measured in the material coordinate system.

If *argu* is equal to 3, VABS will carry out a dehomogenization analysis to compute failure indexes

and strength ratios for each element based on the recovered results of the linear beam theory. The results are stored in *input_file_name.fi*, where the integer number indicating the element number, the following real number is the failure index for the element, and the next real number is the strength ratio for the element. If a failure criterion with clearly identifiable failure modes is used, such as the Hashin failure criterion, VABS will also output the corresponding failure mode for each element at the end of the line for that element. The last line of this output file stores the minimum strength ratio among all the elements, and the smallest element number when this happens. The minimum strength ratio is the safety margin of the cross-section under given global responses.

9 VABS Maintenance and Tech Support

AnalySwift is committed to maintaining and providing tech support for VABS. A discussion group is specifically set up for information exchange related with VABS. *Users are highly encouraged to sign up at https://cdmhub.org/groups/yugroup* to receive most recent news of VABS, ask questions, and share with others. **A technical question should be posted in the discussion group on cdmHUB (https://cdmhub.org/forum) before it will be answered.** A page of VABS FAQ will be constantly updated in the group. Before you ask questions, please do the following:

1. Read the VABS manual carefully, if you have not done so;
2. Check the error message at the end of *input_file_name.ech*;
3. Make sure that you have provided the right input data through *input_file_name.ech*, which is VABS' understanding of your input file;
4. Check the VABS FAQ page on the cdmHUB group;
5. Post your question in the forum on cdmHUB. If your question contains sensitive information, you can ask questions through email.

10 Epilogue

After a period of continuous development ever since year 1989, VABS has reached a level of maturity, and its accuracy has been extensively verified by its developers and users. The performance and robustness of code have been continuously improved based on feedback from its users throughout the world. Although VABS has been designed in such a way that end users do not have to fully understand its theoretical foundation (the details of which are spelled out in VABS related publications), further questions are inevitable because VABS represents a new paradigm to analyze composite beams which is drastically different from other methods. Nevertheless, it should be clear that VABS is the best available code for engineers to design and analyze composite slender structures using beam models.

References

[1] D. H. Hodges, A. R. Atılgan, C. E. S. Cesnik, and M. V. Fulton. On a simplified strain energy function for geometrically nonlinear behaviour of anisotropic beams. *Composites Engineering*, 2(5 – 7):513 – 526, 1992.

[2] W. Yu, D. H. Hodges, V. V. Volovoi, and C. E. S. Cesnik. On Timoshenko-like modeling of initially curved and twisted composite beams. *International Journal of Solids and Structures*, 39(19):5101 – 5121, 2002.

[3] W. Yu, V. V. Volovoi, D. H. Hodges, and X. Hong. Validation of the variational asymptotic beam sectional analysis. *AIAA Journal*, 40(10):2105 – 2113, Oct. 2002.

[4] W. Yu and D. H. Hodges. Elasticity solutions versus asymptotic sectional analysis of homogeneous, isotropic, prismatic beams. *Journal of Applied Mechanics*, 71(1):15 – 23, 2004.

[5] W. Yu and D. H. Hodges. Generalized Timoshenko theory of the variational asymptotic beam sectional analysis. *Journal of the American Helicopter Society*, 50(1):46 – 55, 2005.

[6] W. Yu, D. H. Hodges, V. V. Volovoi, and D. F. Eduardo. A generalized Vlasov theory of composite beams. *Thin-Walled Structures*, 43(9):1493 –1511, 2005.

[7] Dewey H. Hodges. *Nonlinear Composite Beam Theory for Engineers*. AIAA, Washington DC, 2006.

[8] W. Yu. Efficient high-fidelity simulation of multibody systems with composite dimensionally reducible components. *Journal of the American Helicopter Society*, 52(1):49–57, 2007.

[9] D. H. Hodges and W. Yu. A rigorous, engineering-friendly approach for modeling realistic, composite rotor blades. *Wind Energy*, 10(2):179–193, 2007.

[10] S. Roy, W. Yu, and D. Han. An asymptotically correct classical model for smart beams. *International Journal of Solids and Structures*, 44(25-26):8424–8439, 2007.

[11] D. Han, W. Yu, and S. Roy. A geometrically exact active beam theory for multibody dynamics simulation. *Smart Materials and Structures*, 17(4):1136–1147, 2007.

[12] M. A. Neto, W. Yu, and R. P. Leal. Generalized timoshenko modeling of composite beam structures: Sensitivity analysis and optimal design. *Engineering Optimization*, 40(10):891–905, 2008.

[13] S. Roy and W. Yu. Dimensional reduction of an end-electroded piezoelectric composite rod. *European Journal of Mechanics - A/Solids*, 28(2):368–376, 2009.

[14] M. A. Neto, W. Yu, and S. Roy. Two finite elements for general composite beams with piezoelectric actuators and sensors. *Finite Elements in Analysis and Design*, 45(5):295–304, 2009.

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

[16] Q. Wang and W. Yu. Variational asymptotic modeling of the thermal problem of composite beams. *Composite Structures*, 89:1503–1511, 2011.

[17] Q. Wang and W. Yu. Asymptotic multiphysics modeling of composite slender structures. *Smart Materials and Structures*, 21, 2012. Article 035002.

[18] W. Yu, D. H. Hodges, and J. C. Ho. Variational asymptotic beam sectional analysis - an updated version. *International Journal of Engineering Science*, 59:40–64, 2012.

[19] Q. Wang and W. Yu. A refined model for thermoelastic analysis of initially curved and twisted composite beams. *Engineering Structures*, 48:233–244, 2013.

[20] J. C. Ho, W. Yu, and D. H. Hodges. Proper inclusion of interiorly applied loads with beam theory. *Journal of Applied Mechanics*, 80, 2013. Article 011007.

[21] F. Jiang, W. Yu, and D. H. Hodges. Analytical modeling of trapeze and poynting effects of initially twisted beams. *Journal of Applied Mechanics*, 82, 2015. Article 061003.

[22] F. Jiang and W. Yu. Non-linear variational asymptotic sectional analysis of hyperelastic beams. *AIAA Journal*, 54:679–690, 2016.

[23] F. Jiang and W. Yu. Damage analysis by physically nonlinear composite beam theory. *Composite Structures*, 182:652–665, 2017.

[24] F. Jiang, A. Deo, and W. Yu. A composite beam theory for modeling nonlinear shear behavior. *Engineering Structures*, 155:73–90, 2018.

[25] V. L. Berdichevsky. Variational-asymptotic method of constructing a theory of shells. *PMM*, 43(4):664 – 687, 1979.