

VABS Manual for Users*

Wenbin Yu[†]

March 12, 2011

1 Introduction

VABS (*Variational Asymptotic Beam Sectional Analysis*) is a code implementing the various beam theories^{1,2,3,4,5,6,7,8} 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 a powerful mathematical method, the variational asymptotic method.⁹ VABS takes a finite element mesh of the cross section including all the details of geometry and material as inputs to calculate the sectional properties including structural properties and inertial properties. These properties are needed for the 1D beam analysis to predict the global behavior of the slender structure. The 3D pointwise displacement/strain/stress distribution within the structure can also be recovered 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,¹⁰ 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 [11]. The original version of VABS was a research code written in Fortran 77 by Prof. Cesnik. Although the current version of VABS is a completely rewritten code, it is important to note that the original version laid an able foundation 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

*VABS is copyrighted and commercialized by Utah State University Technology Commercialization Office and Georgia Institute of Technology Research Cooperation. All rights reserved.

[†]Associate Professor, Department of Mechanical and Aerospace, Engineering, Utah State University, Logan, Utah 84322-4130.

practicing engineers. The code was rewritten from scratch in Fortran 90/95, with all unnecessary restrictions eliminated, and the computing and memory efficiency greatly enhanced. At the same time, Prof. Censik was continuing his work on VABS for piezoelectric materials at MIT and later at University of Michigan. And Prof. Yu continues his work on VABS for multiphysics modeling at Utah State University. For this reason, there are two variants of VABS: the Georgia Tech/Utah State VABS, released and maintained by Profs. Yu and Hodges, and UM/VABS, released and maintained by Prof. Censik. From henceforth in this manual the term VABS will refer only to the Georgia Tech/Utah State VABS, and what follows is only applicable to this code. Many researchers and engineers all over the world are actively using VABS which is becoming the standard tool for design and analysis of composite slender structures such as helicopter rotor blades, wind turbine blades, high aspect ratio wings, composite bridges, and other slender structural components.

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 generalized mass matrix, and neutral axes, and significant reduction of the computing time for large size problems.

3 What is New in VABS 3.4

The current version of VABS is VABS 3.4. The main new feature is for 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.dat*. A 64-bit version of VABS is also available since VABS 3.4.

4 What is New in VABS 3.3

The current version of VABS is 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 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 calculate just one number for θ_1 for each element and programs flows better with the newly introduced layer definitions.

5 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.¹⁰ 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 3.2
 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 criterion, values expressed in the material coordinate system are necessary (VABS 3.0).
 3. Handles isotropic, orthotropic, and anisotropic material differently. Previous versions treat all materials as orthotropic and 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 dimensional 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 linear beam theory (VABS 3.2.4).
- As far as the quality of the code concerned, VABS 3.2

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. *For a realistic blade section with 200,000 DOFs, VABS 3.0 runs 2 minutes and 2 seconds while VABS 2.1.1 runs 7 minutes 35 seconds on a Dell XPS M1330 laptop.* (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. A VABS manual for developers can be requested separately (VABS 3.0).
5. Has more thorough and informative error handling (VABS 3.0).
6. Uses a simplified license mechanism (VABS 3.2.1)

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. Also starting from VABS III, an limited version of VABS, *VABS Lite*, is provided as a *freeware* for interested users such as students or engineers with limited resources who are not using VABS for real industrial problems.

6 Why VABS Lite

Since January 2006, VABS has been commercialized by Utah State University along with Georgia Institute of Technology. This creates difficulty for some VABS users, such as students using VABS for their studies or engineers using VABS for not-work-related problems. To meet this need, we are committed to develop VABS Lite for each new release as *freeware* available to all interested users. VABS Lite is a limited version of VABS which is capable of classical and Timoshenko modeling of prismatic beams. VABS Lite can model cross section with up to 500 nodes and whatever number of elements formed by the limited number of nodes.

7 VABS Features

Hence along with the features of previous versions, the most recent version of VABS, VABS 3.3, has the following features:

1. It is a highly modularized code written in Fortran 90/95. 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 Fortran 90/95 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 rotor blades with hundreds of layers can be easily handled by 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.

8 VABS Functionalities

VABS is a versatile cross-sectional analysis tool and can obtain all the useful information related with beam sections. Specifically, VABS 3.2 has the following functionalities:

1. *Calculate the 6×6 cross-sectional mass matrix, written in terms of the mass per unit length, and the first and second distributed mass moments of inertia about the three axes.* Based on

the information provided by the mass matrix, VABS calculates the cross-sectional center of mass, the inertial principal axes, the principal mass moments of inertia, and the mass-weighted radius of gyration.

2. *Construct the classical model, represented by a 4×4 stiffness matrix* for prismatic or initially curved/twisted composite beams with normal or oblique cross sections. Based on the classical stiffness model, VABS can calculate the location of neutral axes, or so-called tension center.
3. *Construct the generalized Timoshenko model, represented by a 6×6 stiffness matrix* for prismatic or initially curved/twisted composite beams with normal cross sections. From the generalized Timoshenko model, VABS can calculate the shear center (or elastic axis) location for arbitrary sections.
4. *Construct the generalized Vlasov model, represented by a 5×5 stiffness matrix* for prismatic or initially curved/twisted composite beams with normal cross sections, which is important for thin-walled beams with open sections.
5. *Incorporate the trapeze effect, a nonlinear effect in the 1D constitutive law that stems from 3D geometrical nonlinearity, a very important effect for beams under large centrifugal forces.* The composite beam could be either prismatic or initially twisted and curved.
6. *Recover 3D pointwise displacement, strain, and stress fields for the global behavior of a 1D global beam analysis using the classical model, the generalized Timoshenko model, or the generalized Vlasov model.* Multiple recovery runs can be performed for different inputs of stress resultants without repeating the cross-sectional analysis. The recovered stress/strain fields are evaluated both at the nodal positions and Gaussian points. They are expressed in both the material coordinate system and the beam coordinate system.
7. *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 dimensional reduction for the 3D conduction problem, which can be requested separately.

Note only a subset of aforementioned functionalities are enabled in VABS Lite including classical model and generalized Timoshenko model for prismatic beams with normal cross sections and the corresponding recovery capabilities.

9 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 , where x_1 is along the beam axis and x_2 and x_3 are 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 leading-edge to the trailing edge and points to the direction of the leading edge, and x_3 can be determined

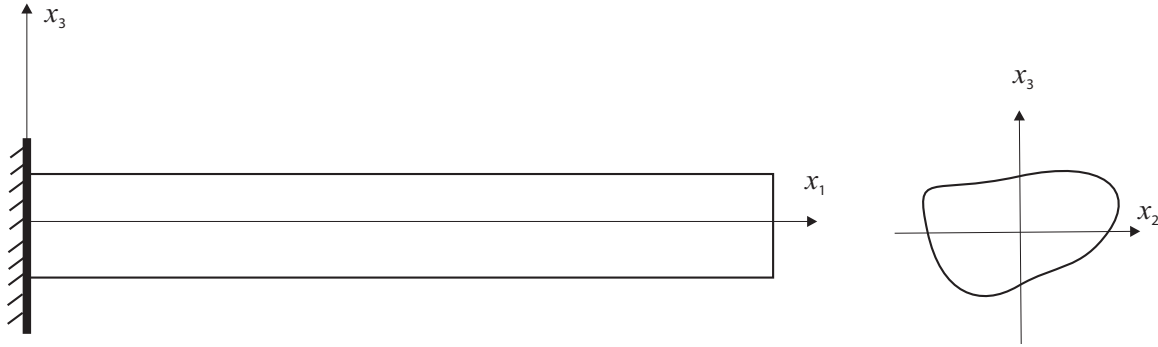


Figure 1: VABS beam coordinate system

by the right hand rule. Usually 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, such as the mass center, centroid, 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. [10].

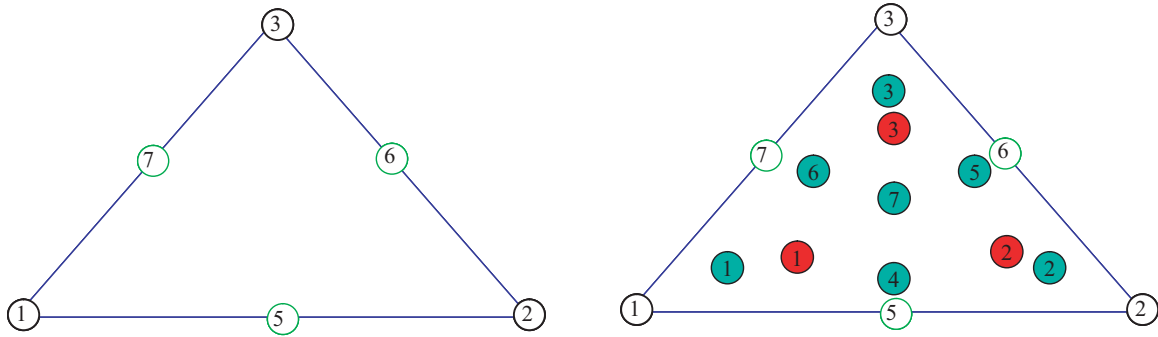


Figure 2: VABS triangular element node numbering and corresponding integration schemes

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 Gaussian integration points. Gaussian integration schemes for different orders of the two types of elements

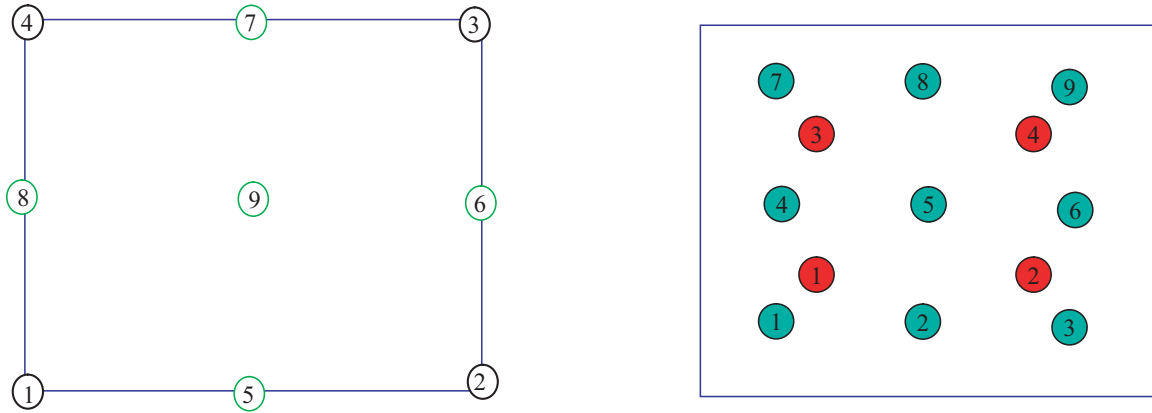


Figure 3: VABS quadrilateral element node numbering and corresponding integration schemes

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 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 essential for applying failure criteria.

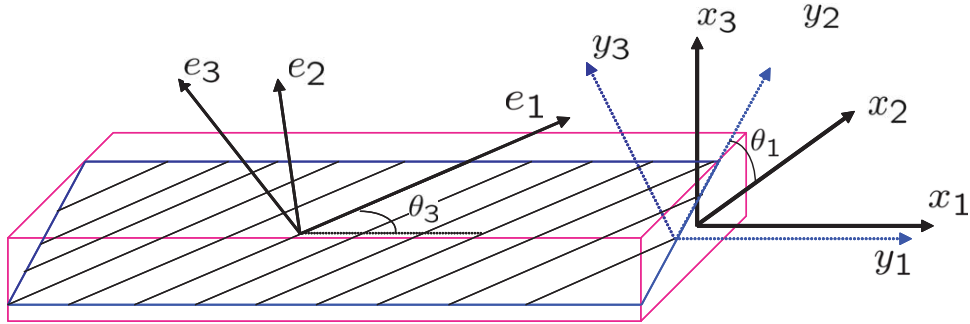


Figure 4: VABS layup convention

VABS allows users to use various kinds of units. However, it is necessary to be absolutely consistent in the choice of units to avoid errors. 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 the slug = lb-sec²/ft; if inches are used for length along with pounds for force, then the unit of mass must be lb-sec²/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

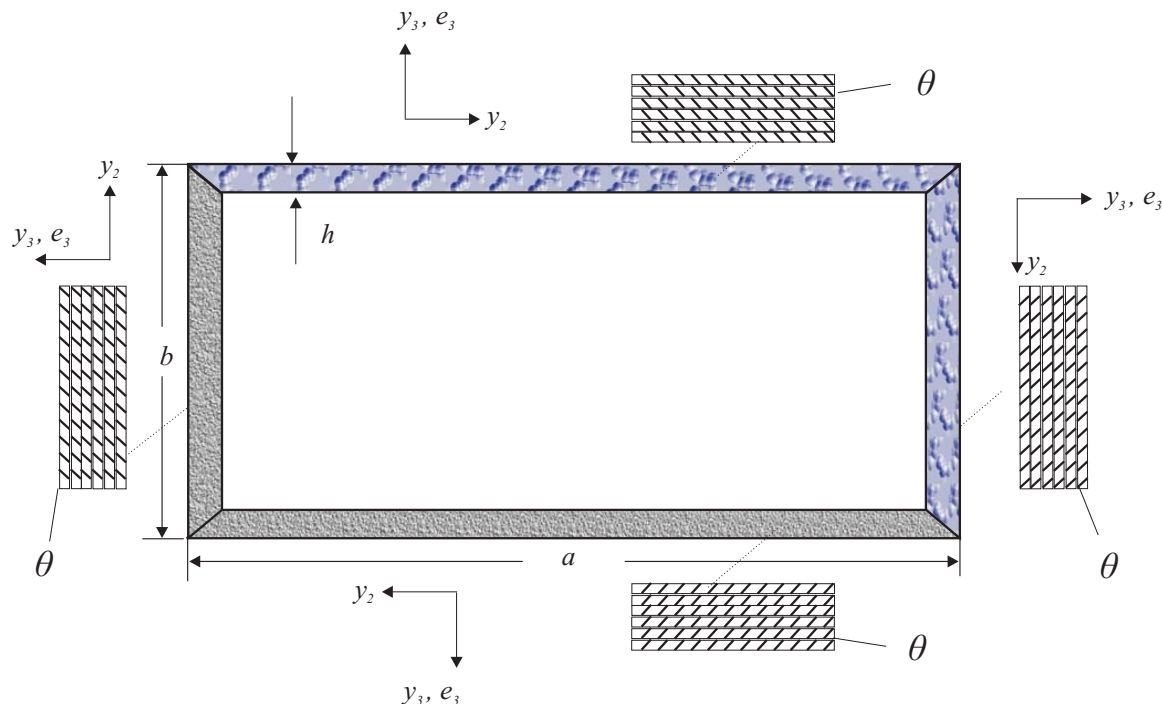


Figure 5: VABS layup convention for a box-beam

the right-hand sense about x_1 by the amount $0 \leq \theta_1 \leq 360^\circ$. Then, the ply coordinate system (y_1, y_2, y_3) is rotated about y_3 in the right-hand sense by the amount $-90^\circ \leq \theta_3 \leq 90^\circ$ to form the material system (e_1, e_2, e_3) , the range of θ_3 being same as is commonly defined in the field of the 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: $\theta_1 = 0^\circ$; the left wall: $\theta_1 = 90^\circ$; the lower wall: $\theta_1 = 180^\circ$; the right wall: $\theta_1 = 270^\circ$. 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.

10 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 familiar environment, to understand the meaning of the input data.

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 three flags arranged as: “*Timoshenko_flag recover_flag thermal_flag*”. The first flag can be only 1 or 0 and the second flag can be 0, 1, or 2. Their uses are explained in the following:

- If *Timoshenko_flag* is 1, VABS will construct both the classical model and the generalized Timoshenko model. If it is 0, it will only construct the classical model.
- If *recover_flag* is 0, VABS will carry out the constitutive modeling. If *recover_flag* is 1, VABS will recover the 3D displacement/strain/stress fields within the cross section based on the *non-linear beam theory*. If *recover_flag* is 2, VABS will recover the 3D displacement/strain/stress fields within the cross section based on the *linear beam theory*. To carry out the 3D recovery, one should have already run VABS once to obtain the required stiffness models including the classical model, the Timoshenko model, or the Vlasov model in the full version with corresponding warping functions stored in *input_file_name.V0* (for the classical warping functions), *input_file_name.V1* (for the warping corrections associated with initially curved and/or twisted beams), and *input_file_name.V1S* (for the warping corrections needed for the generalized Timoshenko or Vlasov model). These warping files are written in binary form and are not directly viewable by the end users. If *reco_flag* is 1 or 2, additional data are needed for inputting the global behavior corresponding to a specific 1D beam analysis, as described later.

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. This flag is automatically set to be 0 in VABS Lite although the user is free to set it as other integer value.

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.

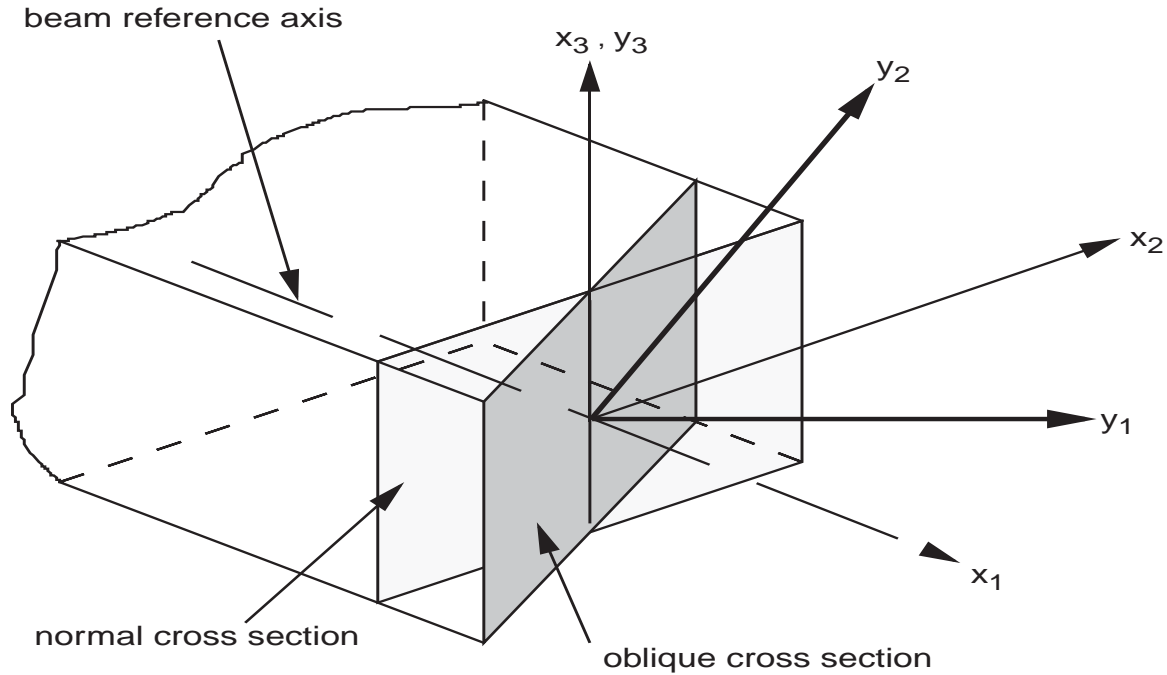


Figure 6: Sketch of an oblique reference cross section

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

These four flags are automatically set to be 0 in VABS Lite although the user is free to set it as any other integer value.

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 x2 x3*,” 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* ($i = 1, 2, \dots, 9$) 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. θ_1 can be calculated at any material point belonging to the element, such as the centroid. **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 one, 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 $\theta_1(2) = 540^\circ$, 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 11 – 15.0

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 ν
 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 ν
 ρ
 α

where α is the coefficient of thermal expansion.

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 generalized Hooke's law for composite materials:

$$\begin{Bmatrix} \epsilon_{11} \\ 2\epsilon_{12} \\ 2\epsilon_{13} \\ \epsilon_{22} \\ 2\epsilon_{23} \\ \epsilon_{33} \end{Bmatrix} = \begin{bmatrix} 1/E_1 & 0 & 0 & -\nu_{21}/E_2 & 0 & -\nu_{31}/E_3 \\ 0 & 1/G_{12} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/G_{13} & 0 & 0 & 0 \\ -\nu_{12}/E_1 & 0 & 0 & 1/E_2 & 0 & -\nu_{32}/E_3 \\ 0 & 0 & 0 & 0 & 1/G_{23} & 0 \\ -\nu_{13}/E_1 & 0 & 0 & -\nu_{23}/E_2 & 0 & 1/E_3 \end{bmatrix} \begin{Bmatrix} \sigma_{11} \\ \sigma_{12} \\ \sigma_{13} \\ \sigma_{22} \\ \sigma_{23} \\ \sigma_{33} \end{Bmatrix}$$

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 α_{11} , α_{22} , α_{33} are the coefficients of thermal expansion 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, 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:

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

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

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

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

$$\begin{array}{cccccc} c_{11} & c_{12} & c_{13} & c_{14} & c_{15} & c_{16} \\ & c_{22} & c_{23} & c_{24} & c_{25} & c_{26} \\ & & c_{33} & c_{34} & c_{35} & c_{36} \\ & & & c_{44} & c_{45} & c_{46} \\ & & & & c_{55} & c_{56} \\ & & & & & c_{66} \\ & & & & & \rho \\ \alpha_{11} & 2\alpha_{12} & 2\alpha_{13} & \alpha_{22} & 2\alpha_{23} & \alpha_{33} \end{array}$$

where α_{ij} , with $i = 1, 2, 3$ and $j = 1, 2, 3$, are the components of the second-order tensor of the thermal expansion coefficients. 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, proper transformation of the material properties is needed.

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 calculating sectional properties, that is, with *recover_flag* being 0.

If *recover_flag* is 1, the user needs to provide additional information obtained from the 1D global beam analysis in the same input file. To carry out recovery 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

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

where $\mathbf{B}_1, \mathbf{B}_2$, and \mathbf{B}_3 are the base vectors of the deformed triad and $\mathbf{b}_1, \mathbf{b}_2$, and \mathbf{b}_3 are the base vectors of the undeformed triad. Details of this definition can be found in Ref. [10]. 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 sectional axial force, M_1 is the sectional torque, M_2 is the sectional bending moment around x_2 , and M_3 is the sectional bending moment around x_3 . The sectional stress and moment resultants are needed for recovering 3D stress and strain distribution within the cross section. For example, if the user only wants to see the pointwise stresses or strains under 1 unit compressive axial force along with 1 unit bending moment around x_2 , the inputs can be arranged as:

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

To carry out recovery based on the generalized Timoshenko model, VABS requires the following data:

$$\begin{array}{ccccccc} u_1 & u_2 & u_3 & & & & \\ C_{11} & C_{12} & C_{13} & & & & \\ C_{21} & C_{22} & C_{23} & & & & \\ C_{31} & C_{32} & C_{33} & & & & \\ F_1 & M_1 & M_2 & M_3 & & & \\ F_2 & F_3 & & & & & \\ f_1 & f_2 & f_3 & m_1 & m_2 & m_3 & \\ f'_1 & f'_2 & f'_3 & m'_1 & m'_2 & m'_3 & \\ f''_1 & f''_2 & f''_3 & m''_1 & m''_2 & m''_3 & \end{array}$$

where the additional data F_2 and F_3 are sectional 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 moment (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 $(\prime) = \frac{\partial}{\partial x_1}$ and $(\prime\prime) = \frac{\partial^2}{\partial x_1^2}$.

To carry out recovery based on the generalized Vlasov model, VABS requires the following data:

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

where $\bar{\gamma}_{11}$ is the beam axial stretching strain measure, $\bar{\kappa}_1$ is the twist measure, $\bar{\kappa}_2$ and $\bar{\kappa}_3$ are the curvature measures around x_2 and x_3 respectively. *It is noted that the global behavior needed for*

recovery should not violate the small strain assumption. Otherwise, you might get some unexpected results. For example, if your transverse shear stiffness is around 2.5 N., then input a shear force resultant of 1 N might be too large as the shear strain is 0.4, which cannot be considered as small.

The input file should be ended with a blank line to avoid any possible incompatibility of different computer 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 60. You are suggested to use a unique extension say *vab* for you to identify such files with VABS. For the convenience of the user to identify mistakes in the input file, all the inputs are echoed in a file named *input_file_name.ech*. Error messages are also written at the end of *input_file_name.ech*.

11 VABS Outputs

Sectional properties obtained by VABS are stored in *input_file_name.K*. Some results are listed as scalars, and some are listed as matrices. Typical scalar results include the mass center, the principal axes, the geometric center, the neutral axes (or tension center), and the elastic axis (or shear center). Typical matrix results include the cross-sectional mass and stiffness matrices of the classical model, the cross-sectional stiffness matrix of the generalized Timoshenko model, the cross-sectional stiffness matrix of the generalized Vlasov model, and the cross-sectional stiffness matrix considering the trapeze effect. The corresponding flexibility matrices are also provided in the output file. These values are all expressed with respect to the beam coordinate system chosen by the user except when the Vlasov flag is equal to 1, the sectional values are expressed with respect to the shear center, although the shear center is still expressed with respect to the user coordinate system. The definitions of these properties are briefly summarized here for the convenience of end users. For more details, please refer to VABS related publications.

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}, x_{m3}) is the location of mass center, i_{22} is the mass moment of inertia about x_2 axis, i_{33} is the mass moment of inertia about x_3 axis, i_{23} is the product of inertia. VABS also calculates the mass matrix at the mass center, and the principal mass moments of inertia and principal inertial axes following standard formulas in textbooks on dynamics. With the knowledge of mass center and principal inertial axes, VABS outputs the mass per unit span, mass moments of inertia about x_1, x_2, x_3 , and mass-weighted radius of gyration, the square of which is the sum of the two in-plane cross-sectional mass moments of inertia (which just happens to be the mass moment of inertia around x_1) divided by the mass per unit span.

The geometric center of the cross section is defined as:

$$Xg_2 = \frac{\langle x_2 \rangle}{\langle 1 \rangle} \quad Xg_3 = \frac{\langle x_3 \rangle}{\langle 1 \rangle}$$

where $\langle \cdot \rangle = \int \int \cdot dx_2 dx_3$. Please noted that only the geometry occupied by a material enters the integration.

The 4×4 stiffness matrix is used to expressed the following constitutive relation for the classical beam model:

$$\begin{Bmatrix} F_1 \\ M_1 \\ M_2 \\ M_3 \end{Bmatrix} = \begin{bmatrix} \bar{S}_{11} & \bar{S}_{12} & \bar{S}_{13} & \bar{S}_{14} \\ \bar{S}_{12} & \bar{S}_{22} & \bar{S}_{23} & \bar{S}_{24} \\ \bar{S}_{13} & \bar{S}_{23} & \bar{S}_{33} & \bar{S}_{34} \\ \bar{S}_{14} & \bar{S}_{24} & \bar{S}_{34} & \bar{S}_{44} \end{bmatrix} \begin{Bmatrix} \bar{\gamma}_{11} \\ \bar{\kappa}_1 \\ \bar{\kappa}_2 \\ \bar{\kappa}_3 \end{Bmatrix} \equiv \bar{S} \begin{Bmatrix} \bar{\gamma}_{11} \\ \bar{\kappa}_1 \\ \bar{\kappa}_2 \\ \bar{\kappa}_3 \end{Bmatrix}$$

The neutral axes or the tension center (Xt_2, Xt_3) is defined as the point of the cross section through which no bending deformation will occur when an axial force applied at this point. The stiffness constants \bar{S}_{ij} depend on the choice of the beam coordinate system, initial curvature/twist, as well as the geometry and material of the cross section. For prismatic beams made of isotropic material with the beam reference line located at the tension center and x_2 and x_3 aligned with the principal bending axes, the stiffness matrix is a diagonal matrix as follows:

$$\begin{bmatrix} EA & 0 & 0 & 0 \\ 0 & GJ & 0 & 0 \\ 0 & 0 & EI_2 & 0 \\ 0 & 0 & 0 & EI_3 \end{bmatrix}$$

the diagonal terms of which are the well known engineering beam constants. For prismatic beams made of isotropic material with the beam reference line not at the tension center and x_2 and x_3 not aligned with the principal bending axes, the stiffness matrix will look like:

$$\begin{bmatrix} EA & 0 & S_3 & -S_2 \\ 0 & GJ & 0 & 0 \\ S_3 & 0 & EI_2 & -EI_{23} \\ -S_2 & 0 & -EI_{23} & EI_3 \end{bmatrix}$$

with $S_\alpha = \langle Ex_\alpha \rangle$ and $EI_{23} = \langle Ex_2x_3 \rangle$.

If *thermal_flag* is equal to 3, VABS also output 1D nonmechanical stress resultants (\bar{N}_T , a 4×1 matrix) and thermal strains ($\bar{\epsilon}_T$, a 4×1 matrix) due to temperature for the classical beam model as

$$\begin{Bmatrix} F_1 \\ M_1 \\ M_2 \\ M_3 \end{Bmatrix} = \bar{S} \begin{Bmatrix} \bar{\gamma}_{11} \\ \bar{\kappa}_1 \\ \bar{\kappa}_2 \\ \bar{\kappa}_3 \end{Bmatrix} - \bar{N}_T$$

and $\bar{\epsilon}_T = \bar{S}^{-1} \bar{N}_T$.

The 6×6 stiffness matrix is used to expressed the following constitutive relation for the generalized Timoshenko model:

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

where $2\gamma_{12}$ and $2\gamma_{13}$ are the engineering transverse shear strains. The generalized shear center or elastic axis (X_{S2}, X_{S3}) defined as the point of the cross section through which no torsion will occur directly due to the sectional transverse shear forces. It is noted that the barred quantities $\bar{\gamma}_{11}, \bar{\kappa}_1, \bar{\kappa}_2, \bar{\kappa}_3$ relate to their unbarred counterparts as $(\bar{\cdot}) = (\cdot)|_{2\gamma_{12}=2\gamma_{13}=0}$.

For initially curved and twisted beams, there are two 4×4 stiffness matrix for the classical model. The first classical stiffness matrix corresponds to the zeroth-order strain energy. The second classical stiffness matrix (correct up to the second order) corresponds to the second-order strain energy with respect to the small parameter related with initial twist/curvatures.

The 5×5 stiffness matrix is used to expressed the following constitutive relation for the generalized Vlasov model:

$$\begin{pmatrix} F_1 \\ M_1 \\ M_2 \\ M_3 \\ M_\omega \end{pmatrix} = \begin{bmatrix} \hat{S}_{11} & \hat{S}_{12} & \hat{S}_{13} & \hat{S}_{14} & \hat{S}_{15} \\ \hat{S}_{12} & \hat{S}_{22} & \hat{S}_{23} & \hat{S}_{24} & \hat{S}_{25} \\ \hat{S}_{13} & \hat{S}_{23} & \hat{S}_{33} & \hat{S}_{34} & \hat{S}_{35} \\ \hat{S}_{14} & \hat{S}_{24} & \hat{S}_{34} & \hat{S}_{44} & \hat{S}_{45} \\ \hat{S}_{15} & \hat{S}_{25} & \hat{S}_{35} & \hat{S}_{45} & \hat{S}_{55} \end{bmatrix} \begin{pmatrix} \bar{\gamma}_{11} \\ \bar{\kappa}_1 \\ \bar{\kappa}_2 \\ \bar{\kappa}_3 \\ \bar{\kappa}'_1 \end{pmatrix}$$

where M_ω is so-called bi-moment corresponding to the twist rate κ'_1 . \hat{S}_{55} is the commonly called “restrained warping” constant, although the complete 5×5 matrix is needed for composite beams with significant restrained warping effects. For thin-walled beams with open sections, it is meaningful to have a generalized Vlasov model based on choosing shear center as its reference. Hence to obtain a generalized 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. It is noted that bars or hats are put over the stiffness constants for different models to indicate the fact that they are not necessarily the same.

The trapeze effect is important for torsionally soft rotating beams. The most general form of a

1D constitutive model including trapeze effect can be represented in the following strain energy

$$\mathcal{U} = \frac{1}{2} \begin{Bmatrix} \gamma_{11} \\ 2\gamma_{12} \\ 2\gamma_{13} \\ \kappa_1 \\ \kappa_2 \\ \kappa_3 \end{Bmatrix}^T \begin{bmatrix} S_{11} & S_{12} & S_{13} & S_{14} & S_{15} & S_{16} \\ S_{12} & S_{22} & S_{23} & S_{24} & S_{25} & S_{26} \\ S_{13} & S_{23} & S_{33} & S_{34} & S_{35} & S_{36} \\ S_{14} & S_{24} & S_{34} & S_{44} & S_{45} & S_{46} \\ S_{15} & S_{25} & S_{35} & S_{45} & S_{55} & S_{56} \\ S_{16} & S_{26} & S_{36} & S_{46} & S_{56} & S_{66} \end{bmatrix} \begin{Bmatrix} \gamma_{11} \\ 2\gamma_{12} \\ 2\gamma_{13} \\ \kappa_1 \\ \kappa_2 \\ \kappa_3 \end{Bmatrix} + \begin{Bmatrix} \bar{\gamma}_{11} \\ \bar{\kappa}_1 \\ \bar{\kappa}_2 \\ \bar{\kappa}_3 \end{Bmatrix}^T (\gamma_{11}A_\gamma + \kappa_1B_{\kappa_1} + \kappa_2C_{\kappa_2} + \kappa_3D_{\kappa_3}) \begin{Bmatrix} \bar{\gamma}_{11} \\ \bar{\kappa}_1 \\ \bar{\kappa}_2 \\ \bar{\kappa}_3 \end{Bmatrix}$$

where $A_\gamma, B_{\kappa_1}, C_{\kappa_2}, D_{\kappa_3}$ can be found in VABS output file if *trapeze_flag* is equal to 1.

The recovered 3D displacement results 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 strain results are stored in *input_file_name.E*. The values are listed for each Gaussian point identified by its location as: “ $x_2 \ x_3 \ \epsilon_{11} \ 2\epsilon_{12} \ 2\epsilon_{13} \ \epsilon_{22} \ 2\epsilon_{23} \ \epsilon_{33}$ ” where ϵ_{ij} are the components of the recovered 3D strain tensor at this Gaussian point expressed in the beam coordinate system.

The recovered 3D stress results are stored in *input_file_name.S*. The values are listed for each Gaussian point identified by its location as: “ $x_2 \ x_3 \ \sigma_{11} \ \sigma_{12} \ \sigma_{13} \ \sigma_{22} \ \sigma_{23} \ \sigma_{33}$ ” where σ_{ij} are the components of the recovered 3D stress tensor at this Gaussian point expressed in the beam coordinate system.

The recovered 3D strain and stress results at each Gaussian point expressed in the material coordinate systems are similarly stored in *input_file_name.EM* and *input_file_name.SM*, respectively. The recovered 3D strain and stress results at each node expressed in the beam coordinate systems are similarly stored in *input_file_name.EN* and *input_file_name.SN*, respectively. The recovered 3D strain and stress results at each node expressed in the material coordinate systems are similarly stored in *input_file_name.EMN* and *input_file_name.SMN*, respectively.

The average of 3D strain and stress results at Gaussian 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 in beam coordinate system, the next six real numbers are the stresses in beam coordinate system, the next six real numbers are strains in material coordinate system, the last six real numbers are the stresses in material coordinate system. This array can be used to facilitate contour plot for visualization.

All these output files are in pure text format and can be opened by any text editor.

12 VABS Installation

VABS is distributed in the form of VABSx.xReleasePcMM-DD-YEAR.zip for Windows operating systems with “x.x” denotes the version number. Sometimes to circumvent some email systems, the extension “zip” is changed to be ”pass”. You just need to change it back to be a zip file once you received it. If you also want to run VABS in other platforms such as Unix/Linux or Mac, please let the author know. Extracting the file VABSx.xReleasePcMM-DD-YEAR.zip to a folder you choose is all you need to do for installation. There are at least three ways to run VABS:

- The safest way to execute VABS is to run *VABSIII inputfile* as a DOS command in Windows. This can be achieved by Click Start, choose Run, and type in “cmd” and click OK. Then use “cd” to enter the right folder where both the VABS executable and the input file is in.
- Drag your input file into the VABS executable. A window appears to run VABS along with the input file. After it is done, the window will disappear and you have new files generated in the folder of the input file. You can check your results in the output files as described previously.
- Assign a unique extension, say *vab*, to your input file, and configure files with this extension to be opened by VABS and to be edited by a text editor. From a folder’s window, click Tools->Folder Options. Once the Folder Options dialog is open, click tab File Types. Click the New icon to Create New Extension dialog, type *vab* and click OK. With the new file type *vab* highlighted in the Registered file types list, click Advanced to open the Edit File Type dialog. Click Change icon to choose the icon you like. Next, click on New, type *open* under Action, then click Browse and point to the VABS executable and click OK. Click on New, type edit under Action, then click Browse to find the executable for text editor you like to use for editing input files and click OK. Click OK and Close to exit the Folder Options dialog. Then you only need to double click the input file, it will launch VABS for you. You can also rightclick an input file and open it with the text editor.

13 VABS Maintenance and Tech Support

Prof. Yu is committed to maintaining and providing tech support for VABS. A Google group is specifically set up for information exchange related with VABS. *Users are highly encouraged to sign up through <http://groups.google.com/group/hifi-comp> to receive most recent news of VABS, ask questions, and share with others. A technical question should be posted in the Google group 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’s understanding of your input file;
4. Check the VABS FAQ page on the Google group;

5. Post your question in the discussion section of the group.

A web site (hifi-comp.com) has also been created to provide more exposure of the codes Prof. Yu has developed including VABS, VAPAS, and VAMUCH through Internet. .

14 Epilogue

After a period of continuous development spanning more than 15 years, 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 beam cross sections.

References

- [1] 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.
- [2] 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.
- [3] 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.
- [4] 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.
- [5] 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.
- [6] D. H. Hodges and W. Yu. A rigorous, engineering-friendly approach for modeling realistic, composite rotor blades. *Wind Energy*, 10(2):179–193, 2007.
- [7] 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.
- [8] 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.
- [9] V. L. Berdichevsky. Variational-asymptotic method of constructing a theory of shells. *PMM*, 43(4):664 – 687, 1979.

- [10] Dewey H. Hodges. *Nonlinear Composite Beam Theory for Engineers*. AIAA, Washington DC, 2006.
- [11] D. H. Hodges, A. R. Atilgan, 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.