SwiftComp™ User Manual

Version 2.1

Wenbin Yu*

March 25, 2022

1 Introduction

SwiftComp^{TM} represents a general-purpose approach for computing effective properties (aka constitutive modeling) of composite materials and structures. Here composite materials and structures refer to those materials and structures featuring anisotropy and heterogeneity, not just the traditional fiber reinforced polymers or unidirectional laminates. SwiftComp^{TM} can be used independently for virtual testing composite materials and structures or as a plugin to power conventional finite element analysis (FEA) codes with efficient high-fidelity multiscale modeling for such materials and structures. SwiftComp^{TM} implements Mechanics of Structure Genome (MSG), a unique multiscale modeling approach based on the concept of Structure Gene (SG), to capture both anisotropy and heterogeneity of composites at the microscopic scale or other scales of user's interest. MSG unifies micromechanics and structural mechanics to provide a single theory to model all types of composite materials and structures. SwiftComp^{TM} enables engineers to analyze composite materials and structures similarly to metals, capturing details as needed and affordable.

To facilitate the use of SwiftComp^{TM}, several graphic user interfaces have been developed including Gmsh4SC, TexGen4SC, Abaqus-SwiftComp GUI, Ansys-SwiftComp GUI, and Nastran-SwiftComp GUI. Instructions for using SwiftComp^{TM} through these GUIs are given in user manuals accompanying the corresponding GUIs. This manual will provide an introduction to MSG, the history and functionalities of SwiftComp^{TM}, conventions, inputs, outputs, maintenance, and tech support for more advanced SwiftComp users.

*Professor, School of Aeronautics and Astronautics, Purdue University; Director, Composites Design and Manufacturing HUB (cdmHUB); CTO, AnalySwift LLC.

2 SwiftComp™ History

SwiftComp™ is a culmination of prior work on composite structures and materials implemented in three different codes including VABS, VAPAS, and VAMUCH developed by Prof. Wenbin Yu and his coworkers. The VABS code was developed for composite beam modeling during Prof. Yu's PhD study at Georgia Tech under the supervision of Prof. Dewey Hodges. VABS was later significantly enhanced at Utah State through its affiliation with Georgia Tech's rotorcraft center and commercialized by Utah State through AnalySwift LLC. The VAPAS code was developed to model composite laminated plates and shells at Georgia Tech and VAPAS was also later enhanced at Utah State. VAMUCH, also called SwiftComp Micromechanics™, is a general-purpose micromechanics code for homogenization and dehomogenization of periodic, heterogeneous materials developed at Utah State. In year 2012, Prof. Yu introduced the representative structural element (RSE) concept to unify structural mechanics and micromechanics for multiscale constitutive modeling of composites [1]. RSE concept was later renamed as SG to emphasize its role in filling the gap between materials genome and structural analysis [2]. The founding paper of MSG was published in year 2016 [3] featuring a general geometrical nonlinear formulation, which was simplified to linear problems later in [4]. To implement MSG, we started the development of SwiftComp™ in April 2014 at Purdue University. SwiftComp™ is a single code which can reproduce all of the functionalities in VABS, VAPAS, and VAMUCH, as well as many other capabilities not found in any of these three codes. SwiftComp™ can reproduce VABS for composite beams made of uniform cross-sections (see the right figure of Figure 4), VAPAS for composite laminated plates and shells (see the right figure of Figure 5), and VAMUCH for 3D periodic heterogeneous materials (see Figure 3). However, currently not all VABS capabilities are available in SwiftComp™ and VABS is still maintained as a separate code for cross-sectional analysis of composite beams while VAPAS and VAMUCH are superseded by SwiftComp™. SwiftComp™ has many more functionalities not available in the previous three codes such as slender structures with spanwise heterogeneities, plates and shells with in-plane heterogeneity, partially periodic structures and materials, aperiodic structures and materials, etc.

3 SwiftComp™ Functionalities

Fundamentally speaking, SwiftComp™ takes the geometry and material characteristics of an SG described using a finite element mesh as the input and computes the effective properties for the macroscopic analysis. This process is commonly called *homogenization*. SwiftComp™ can also compute the local fields within the SG based on global behavior obtained from the macroscopic analysis. This process is commonly called *dehomogenization*, constantly neglected in some multi-scale modeling approaches. Note that SwiftComp™ is not limited to structural modeling, it can be used to perform multiphysics homogenization and dehomogenization of materials and structures responsive to thermal, mechanical, electric, and magnetic fields.

3.1 Alpha Version

The alpha version of SwiftComp™ can perform the constitutive modeling corresponding to the classical structural models including the Euler-Bernoulli beam model, Kirchhoff-Love plate/shell

model, and 3D Cauchy continuum model.

3.2 Version 1.0

Starting SwiftComp^{TM} 1.0, two versions of SwiftComp^{TM} is possible: SwiftComp^{TM} Standard and SwiftComp^{TM} Professional. In SwiftComp^{TM} Professional, a direct sparse solver is used to deal with big models which could have as many as millions of degrees of freedom. A parallel edition is also available for SwiftComp^{TM} Professional. For a problem that SwiftComp^{TM} Professional runs more than a few minutes, it is better to use the parallel edition as it can exploit multiple cores which are readily available on most computers nowadays. In the standard version, Prof. Sloan's method[5] is used to provide the renumbering of the finite element mesh, standard skyline storage is used along with a regular direct linear solver. To simplify the maintenance of the code, the later official released versions use the parallel version with a direct sparse solver.

3.2.1 Version 1.1

The pointwise anisotropic heterogeneity is enabled in SwiftComp^{TM} 1.1. The SG can contain phases with general anisotropic materials with material properties given in material coordinates which could be different from the local coordinate system and defined for each element. First, an elemental coordinate system is defined by three points for each element, then the material coordinate system can be defined as a simple rotation around one of the axis of the elemental coordinate element. The first capability (orientation described using an elemental coordinate system) is found applications in woven components and short fiber reinforced composites. The second capability (orientation described using a rotation angle) is found applications in composite laminates.

3.2.2 Version 1.2

SwiftComp^{TM} 1.2 implements the capability to deal with aperiodic, or partially periodic SGs, or periodic SGs without periodic nodes on the boundary surfaces.

3.2.3 Version 1.3

SwiftComp^{TM} 1.3 implements the capability to predict initial failure strengths according to a given failure criterion, failure envelopes for giving two load directions, and failure indexes and strength ratios for materials and structures subject to an arbitrary loading.

3.2.4 Version 1.4

SwiftComp^{TM} 1.4 implements homogenization, dehomogenization, initial failure analysis according to the Timoshenko beam model.

3.2.5 Version 1.5

SwiftComp^{TM} 1.5 implements a more efficient and robust way for predicting failure envelopes and also a few minor bugs were fixed.

3.2.6 Version 2.0

SwiftComp^{TM} 2.0 implements thermoviscoelastic analysis capabilities, temperature change within the SG, and wedge element. Also a few minor bugs were fixed.

3.2.7 Version 2.1

SwiftComp^{TM} 2.1 perform constitutive modeling (homogenization and dehomogenization) of a block of 3D elements to be a 3D 8-node or 20-node element.

4 Structures and Structural Models

To understand the meaning of SG, we need to define what is a structure. Structure and material often appear together and sometimes are used interchangeably. The main reason is that the difference between structure and material is not clear. The interchangeable use of these two terms starts from the first solid mechanics course, usually called either *Strength of Materials* or *Mechanics of Materials*, mainly covering stress analysis of various structures including rods, shafts, columns, beams, pressure vessels, etc. With the extensive penetration of composites in industry and the increasing capabilities of fabricating materials with complex microstructures, the difference between structure and material becomes even more elusive.

One possible difference is the presence of boundary: the material itself has no boundaries, but rather may be considered as a point in a structure according to the continuum hypothesis. Another difference is that we describe a material using material properties such as Young's moduli, Poisson's ratios, coefficients of thermal expansion (CTEs), yielding limits, ultimate strengths, etc. These properties are intrinsic to the material, and will not change for different structures made of the same material. For example, if a linear elastic material has Young's modulus equal to 73 GPa, it will always remain this value no matter whether it is used to make a shaft or a pressure vessel, subject to a force of 10 N or 10 kN. It is noted that for nonlinear behavior, the material properties could depend on the current stress or strain state which the material is experiencing but these properties are genetic (or intrinsic) to the material which means even if the material is not experiencing any stress or strain right now, the same constitutive relations exist.

On the other hand, a structure is a solid body made of one or more materials with clearly defined interactions with its external environment through boundary conditions, applied loads, temperature, moisture, etc. The behavior of a structure is usually described using displacements, strains, stresses, natural frequencies, buckling loads, failure, etc. Structures can be categorized in terms of their external geometry. If the three dimensions of a structure are of similar size, it is called a 3D structure (Figure 1 a). If one dimension of the structure is much *smaller* than the two other dimensions, the structure is called as a plate (Figure 1 b) or shell (Figure 1 c) depending on whether it is flat or curved along the two large dimensions. Usually the small dimension is called thickness and the two large dimensions are called the in-plane directions. A reference surface can be defined for a plate or shell using the two in-plane coordinates. If one dimension of the structure is much *larger* than two other dimensions, the structure is called as a beam (Figure 1 d). Usually the

large dimension is called the length, span, or axis of the beam and a reference line can be defined for a beam using the axial coordinate. The reference line can be as general as a spatial curve which is the case for initially curved and twisted beams. The two small dimensions are commonly called the cross-section for typical beam-like structures. As far as the mathematical modeling is concerned, this classification only considers the external geometry and the internal construction of these structures can be arbitrary. For example, a porous material with holes in the body can be modeled as a 3D solid, a sandwich flat panel with a honeycomb core can be modeled as a plate, a high aspect ratio wing of aircraft can be modeled as a beam with no clearly defined cross-sections. In this sense, all engineering structural systems, despite its complexity, can be considered as formed by using a combination of structural components in terms of 3D structures, plates/shells, and/or beams as shown in Figure 1 with possible complex internal constructions.

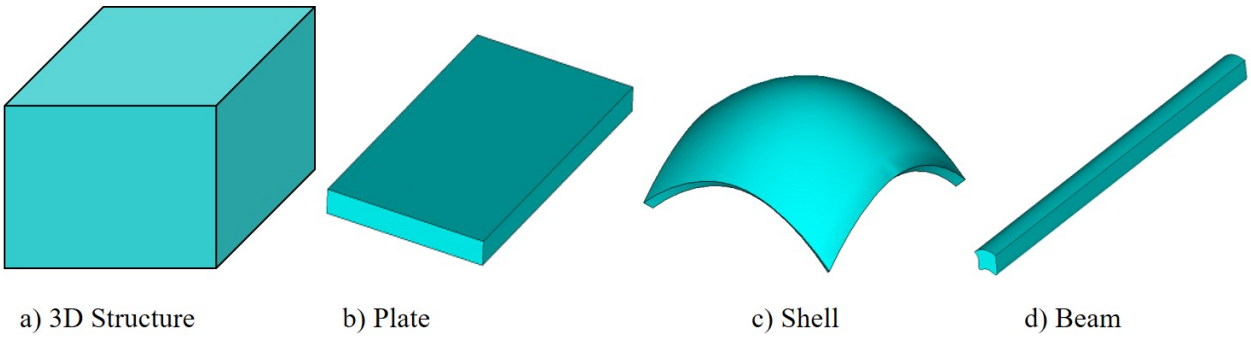

Figure 1: Typical structural components.

How to predict displacements, strains, and stresses within structures using mathematical models belongs to the long standing branch of applied mechanics call *structural mechanics*. Although there are historical and practical reasons for classifying structures as 3D structures, plates/shells, and beams, different names for different structures have also motivated us to construct different mathematical models for modeling different structures. Let us use x_1, x_2, x_3 to denote the global Cartesian coordinate system. Beams, plates, and shells are also collectively called *dimensionally reducible structures* because it is possible for us to eliminate the small dimension(s) to create dimensionally reduced models (1D or 2D models) for these structures [6]. More specifically, we can create one-dimensional (1D) models (also called beam models) for beam-like structures. The field variables of beam models are functions of a single coordinate x_1 describing the beam reference line. We can also create two-dimensional (2D) models (also called plate/shell models) for plate/shell-like structures. The field variables of plate/shell models are functions of x_1 and x_2, the two in-plane coordinates describing the plate/shell reference surface. Here, the notation of 1D, 2D, or 3D refers to the number of coordinates needed to describe the analysis domain. It does not correspond to the dimensionality of the behavior. For example, a 1D beam model can describe the displacements and the rotations of the structure in three directions in space.

Each model contains three sets of equations including kinematics, kinetics, and constitutive relations, among which only the constitutive relations change for different materials used to make the structure. For composites, the stiffness matrix could be fully populated. The task to compute the

constitutive relations is called *constitutive modeling*. 3D properties (*e.g.* Young's moduli, Poisson's ratios, and shear moduli, etc.) could be obtained either experimentally or through a micromechanics calculation. For isotropic homogeneous structures, 3D material properties are direct inputs for structural analysis using a 3D solid model, and these properties combined with geometric characteristics of the structure can be used for a structural analysis using plate/shell/beam models with the structural stiffness (*e.g.* EA, EI, GJ, etc.) obtained based on apriori assumptions such as the Euler-Bernoulli assumptions for beams. However, such straightforwardness does not exist for structures featuring anisotropy and/or heterogeneity. Many refined assumptions, such as higher-order assumptions, zigzag assumptions, layerwise assumptions, have been introduced to better capture the kinematics along the smaller dimensions of the structure (the thickness of a plate/shell or the cross-section of a beam) in most other approaches.

Structural analyses are routinely carried out using finite element analysis (FEA) codes such as Abaqus, Ansys, and Nastran in terms of 3D solid elements, 2D plate or shell elements, or 1D beam elements (see Figure 2). Each element type is governed by a corresponding structural model which contains kinematics, kinetics, and constitutive relations. The commonly used engineering beam models include the Euler-Bernoulli model and the Timoshenko model. The Euler-Bernoulli model is also commonly called the classical beam model. The commonly used engineering plate/shell models are the Kirchhoff-Love model and the Reissner-Mindlin model. The Kirchhoff-Love model is also commonly called the classical plate/shell model and the Reissner-Mindlin model is also commonly called the first-order shear-deformation plate/shell model. The commonly used engineering 3D model is the Cauchy continuum model which is also commonly called the classical continuum model. Here, we will restrict ourselves to structures made of linear elastic materials to illustrate the basics of each structural model although it is emphasized that MSG has no such restriction. The models for structures featuring geometrical and material nonlinearities can be found in relevant MSG publications.

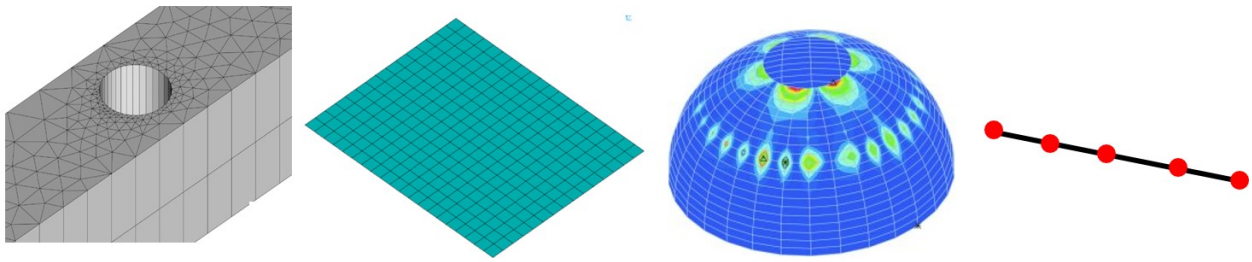

Figure 2: Typical structural elements.

4.1 Cauchy Continuum Model

The kinematics of the Cauchy continuum model contains three displacements (u_1, u_2, u_3) and six strains $(\varepsilon_{11}, \varepsilon_{22}, \varepsilon_{33}, \varepsilon_{23}, \varepsilon_{13}, \varepsilon_{12})$. The strain-displacement relations are given as

$$\begin{aligned} \varepsilon_{11} &= \frac{\partial u_1}{\partial x_1}, \qquad \varepsilon_{22} = \frac{\partial u_2}{\partial x_2}, \qquad \varepsilon_{33} = \frac{\partial u_3}{\partial x_3} \\ 2\varepsilon_{23} &= \frac{\partial u_2}{\partial x_3} + \frac{\partial u_3}{\partial x_2}, \qquad 2\varepsilon_{13} = \frac{\partial u_1}{\partial x_3} + \frac{\partial u_3}{\partial x_1}, \qquad 2\varepsilon_{12} = \frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1} \end{aligned} \tag{1}$$

The kinetics of the Cauchy continuum model is described using six stresses $(\sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{23}, \sigma_{13}, \sigma_{12})$ which are functions of x_1, x_2, x_3. These stresses are governed by the following equations of equilibrium:

$$\begin{aligned} \frac{\partial \sigma_{11}}{\partial x_1} + \frac{\partial \sigma_{12}}{\partial x_2} + \frac{\partial \sigma_{13}}{\partial x_3} + f_1 &= 0 \\ \frac{\partial \sigma_{12}}{\partial x_1} + \frac{\partial \sigma_{22}}{\partial x_2} + \frac{\partial \sigma_{23}}{\partial x_3} + f_2 &= 0 \\ \frac{\partial \sigma_{13}}{\partial x_1} + \frac{\partial \sigma_{23}}{\partial x_2} + \frac{\partial \sigma_{33}}{\partial x_3} + f_3 &= 0 \end{aligned} \tag{2}$$

where f_1, f_2, f_3 are distributed body forces per unit volume. The constitutive relations of the Cauchy continuum model for the linear elastic behavior are described using the Hooke's law as

$$\begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{12} \end{Bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ C_{12} & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} \\ C_{13} & C_{23} & C_{33} & C_{34} & C_{35} & C_{36} \\ C_{14} & C_{24} & C_{34} & C_{44} & C_{45} & C_{46} \\ C_{15} & C_{25} & C_{35} & C_{45} & C_{55} & C_{56} \\ C_{16} & C_{26} & C_{36} & C_{46} & C_{56} & C_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ 2\varepsilon_{23} \\ 2\varepsilon_{13} \\ 2\varepsilon_{12} \end{Bmatrix} \tag{3}$$

The 6×6 matrix C_{ij} is called the stiffness matrix and its inverse is called the compliance matrix. Note that material properties are usually provided in the so-called material coordinate system which implies that we need to write constitutive relations in the material coordinate system first. However, the kinematics and kinetics are usually formulated in the global coordinate system. A proper transformation according to the tensorial transformation laws is needed to transfer the constitutive relations into the global coordinate system.

For isotropic materials, the constitutive relations can be expressed in terms of the Young's modulus E and Poisson's ratio ν as

$$\begin{Bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ 2\varepsilon_{23} \\ 2\varepsilon_{13} \\ 2\varepsilon_{12} \end{Bmatrix} = \begin{bmatrix} \frac{1}{E} & -\frac{\nu}{E} & -\frac{\nu}{E} & 0 & 0 & 0 \\ -\frac{\nu}{E} & \frac{1}{E} & -\frac{\nu}{E} & 0 & 0 & 0 \\ -\frac{\nu}{E} & -\frac{\nu}{E} & \frac{1}{E} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{2(1+\nu)}{E} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{2(1+\nu)}{E} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{2(1+\nu)}{E} \end{bmatrix} \begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{12} \end{Bmatrix} \tag{4}$$

which can be inverted to obtain the same expression as Eq. (3) with

$$
\begin{aligned}
C_{11} = C_{22} = C_{33} &= \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} \\
C_{12} = C_{13} = C_{23} &= \frac{E\nu}{(1+\nu)(1-2\nu)} \\
C_{44} = C_{55} = C_{66} &= \frac{E}{2(1+\nu)}
\end{aligned} \tag{5}
$$

and all other terms in the stiffness matrix of Eq. (3) are zero.

For orthotropic materials, the constitutive relations can be expressed as

$$
\begin{Bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ 2\varepsilon_{23} \\ 2\varepsilon_{13} \\ 2\varepsilon_{12} \end{Bmatrix} =
\begin{bmatrix}
\frac{1}{E_1} & -\frac{\nu_{21}}{E_2} & -\frac{\nu_{31}}{E_3} & 0 & 0 & 0 \\
-\frac{\nu_{12}}{E_1} & \frac{1}{E_2} & -\frac{\nu_{32}}{E_3} & 0 & 0 & 0 \\
-\frac{\nu_{13}}{E_1} & -\frac{\nu_{23}}{E_2} & \frac{1}{E_3} & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{1}{G_{23}} & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{1}{G_{13}} & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{1}{G_{12}}
\end{bmatrix}
\begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{12} \end{Bmatrix} \tag{6}
$$

where E_1, E_2, E_3 are Young's moduli in three directions, G_{12}, G_{13}, G_{23} are shear moduli in three directions, $\nu_{12}, \nu_{13}, \nu_{23}$ and $\nu_{21}, \nu_{31}, \nu_{32}$ are two set of Poisson's ratios. Usually only $\nu_{12}, \nu_{13}, \nu_{23}$ are given and the other Poisson's ratios are calculated as

$$
\nu_{21} = \nu_{12}E_2/E_1, \qquad \nu_{31} = \nu_{13}E_3/E_1, \qquad \nu_{32} = \nu_{23}E_3/E_2
$$

due to the symmetry of the compliance matrix.

Eq. (6) can be inverted to obtain the same expression as Eq. (3) with

$$
\begin{aligned}
&C_{11} = E_1(1-\nu_{23}\nu_{32})/\Delta, \qquad C_{12} = E_2(\nu_{12}+\nu_{13}\nu_{32})/\Delta, \qquad C_{13} = E_3(\nu_{13}+\nu_{12}\nu_{23})/\Delta, \\
&C_{22} = E_2(1-\nu_{13}\nu_{31})/\Delta, \qquad C_{23} = E_3(\nu_{23}+\nu_{13}\nu_{21})/\Delta, \qquad C_{33} = E_3(1-\nu_{12}\nu_{21})/\Delta, \\
&C_{44} = G_{23}, \qquad C_{55} = G_{13}, \qquad C_{66} = G_{12}
\end{aligned}
$$

with

$$
\Delta = 1 - \nu_{12}\nu_{21} - \nu_{23}\nu_{32} - \nu_{13}\nu_{31} - 2\nu_{21}\nu_{13}\nu_{32}
$$

and all other terms in Eq. (3) are zero.

Eqs. (1), (2), and (3) form a system of 15 equations underpinning the Cauchy continuum model to be solved along with appropriate boundary conditions for 15 unknowns (three displacements, six strains, and six stresses). This model has been implemented in many FEA codes which have 3D solid elements. Kinematics and kinetics remain the same no matter whether the structure is made of metals or composites. Only the constitutive relations will be different, see Eq. (3) for general anisotropic materials, Eq. (4) for isotropic materials, and Eq. (6) for orthotropic materials.

It is worthy to point out that there are two degenerated version of the 3D Cauchy continuum model: plane stress model and plane strain model. Plane stress model assumes that all the out-of-plane stresses vanish. For example, if the plane stress model is formulated in the $x_1 - x_2$ plane, we assume $\sigma_{13} = \sigma_{23} = \sigma_{33} = 0$. Plane strain model assumes that all the out-of-plane strains vanish. For example, if the plane strain model is formulated in the $x_1 - x_2$ plane, we assume $\varepsilon_{13} = \varepsilon_{23} = \varepsilon_{33} = 0$. The kinematics of the plane stress model and the plane strain model remain the same as

$$\varepsilon_{11} = \frac{\partial u_1}{\partial x_1}, \qquad \varepsilon_{22} = \frac{\partial u_2}{\partial x_2}, \qquad 2\varepsilon_{12} = \frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1} \tag{7}$$

The kinetics of the plane stress model and the plane strain model remain the same as

$$\begin{aligned} \frac{\partial \sigma_{11}}{\partial x_1} + \frac{\partial \sigma_{12}}{\partial x_2} + f_1 = 0 \\ \frac{\partial \sigma_{12}}{\partial x_1} + \frac{\partial \sigma_{22}}{\partial x_2} + f_2 = 0 \end{aligned} \tag{8}$$

The constitutive relations of the plane strain model are

$$\begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{Bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{16} \\ C_{12} & C_{22} & C_{26} \\ C_{16} & C_{26} & C_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ 2\varepsilon_{12} \end{Bmatrix} \tag{9}$$

The constitutive relations of the plane stress model are

$$\begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{Bmatrix} = \begin{bmatrix} Q_{11} & Q_{12} & Q_{16} \\ Q_{12} & Q_{22} & Q_{26} \\ Q_{16} & Q_{26} & Q_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ 2\varepsilon_{12} \end{Bmatrix} \tag{10}$$

where Q_{ij} are the so-called plane-stress-reduced stiffnesses which are different from C_{ij}. They are computed by substituting $\sigma_{13} = \sigma_{23} = \sigma_{33} = 0$ into Eq. (3) to obtain $\varepsilon_{33}, 2\varepsilon_{23}, 2\varepsilon_{13}$ in terms of $\varepsilon_{11}, \varepsilon_{22}, 2\varepsilon_{12}$. Then substituting these relations back into Eq. (3) to obtain the relationship in Eq. (10). For isotropic materials, we have

$$Q_{11} = Q_{22} = \frac{E}{1-\nu^2}, \qquad Q_{12} = \frac{E\nu}{1-\nu^2}, \qquad Q_{16} = Q_{26} = 0, \qquad Q_{66} = \frac{E}{2(1+\nu)}$$

which are different from C_{ij} for isotropic materials given in Eq. (5). For anisotropic materials, Q_{ij} expressions can be found in a typical textbook on mechanics of composite materials.

4.2 Kirchhoff-Love Plate/Shell Model

Kirchhoff originally developed the classical plate model for flat panels based on a set of ad hoc assumptions including the transverse normal line being rigid in the thickness direction, perpendicular to the reference surface, and plane stress assumption. Love later extended the same set of assumptions to curved panels to develop the classical shell model. Since both models are based on the same set of assumptions and assume the same model form, we call them collectively as the

Kirchhoff-Love model. Here, we use the plate model for illustrative purpose. The kinematics of the Kirchhoff-Love model contains three displacements (u_1, u_2, u_3) and six strain variables including in-plane strains $(\epsilon_{11}, \epsilon_{22}, \epsilon_{12})$ and curvatures $(\kappa_{11}, \kappa_{22}, \kappa_{12})$. For a plate, the strain-displacement relations are given as

$$\begin{aligned} \epsilon_{11} &= \frac{\partial u_1}{\partial x_1}, \qquad \epsilon_{22} = \frac{\partial u_2}{\partial x_2}, \qquad 2\epsilon_{12} = \frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1} \\ \kappa_{11} &= -\frac{\partial^2 u_3}{\partial x_1^2}, \qquad \kappa_{22} = -\frac{\partial^2 u_3}{\partial x_2^2}, \qquad \kappa_{12} = -\frac{\partial^2 u_3}{\partial x_1 \partial x_2} \end{aligned} \tag{11}$$

The kinetics of the Kirchhoff-Love model contains six stress resultants $(N_{11}, N_{22}, N_{12}, M_{11}, M_{22}, M_{12})$ with N_{11}, N_{22}, N_{12} denoting in-plane forces and M_{11}, M_{22}, M_{12} denoting moments. These kinetic variables are governed by the following three equations of equilibrium

$$\begin{aligned} &\frac{\partial N_{11}}{\partial x_1} + \frac{\partial N_{12}}{\partial x_2} + p_1 = 0 \\ &\frac{\partial N_{21}}{\partial x_1} + \frac{\partial N_{22}}{\partial x_2} + p_2 = 0 \\ &\frac{\partial^2 M_{11}}{\partial {x_1}^2} + \frac{\partial^2 M_{22}}{\partial {x_2}^2} + 2\frac{\partial^2 M_{12}}{\partial x_1 \partial x_2} + \frac{\partial q_2}{\partial x_1} - \frac{\partial q_1}{\partial x_2} + p_3 = 0 \end{aligned} \tag{12}$$

where p_1, p_2, p_3 are equivalent forces and q_1, q_2 are equivalent moments, distributed over the reference surface.

The constitutive relations of the Kirchhoff-Love model can be expressed using the following matrix equation.

$$\begin{Bmatrix} N_{11} \\ N_{22} \\ N_{12} \\ M_{11} \\ M_{22} \\ M_{12} \end{Bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{16} & B_{11} & B_{12} & B_{16} \\ A_{12} & A_{22} & A_{26} & B_{12} & B_{22} & B_{26} \\ A_{16} & A_{26} & A_{66} & B_{16} & B_{26} & B_{66} \\ B_{11} & B_{12} & B_{16} & D_{11} & D_{12} & D_{16} \\ B_{12} & B_{22} & B_{26} & D_{12} & D_{22} & D_{26} \\ B_{16} & B_{26} & B_{66} & D_{16} & D_{26} & D_{66} \end{bmatrix} \begin{Bmatrix} \epsilon_{11} \\ \epsilon_{22} \\ 2\epsilon_{12} \\ \kappa_{11} \\ \kappa_{22} \\ 2\kappa_{12} \end{Bmatrix} \tag{13}$$

This 6×6 matrix is commonly called the plate stiffness matrix and its inverse is called the plate compliance matrix for the Kirchhoff-Love model. If a plate is made of a single isotropic material, and the origin of x_3 is located at the center of the thickness, the constitutive relations can be written as

$$\begin{Bmatrix} N_{11} \\ N_{22} \\ N_{12} \end{Bmatrix} = \frac{Eh}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \begin{Bmatrix} \epsilon_{11} \\ \epsilon_{22} \\ 2\epsilon_{12} \end{Bmatrix}, \qquad \begin{Bmatrix} M_{11} \\ M_{22} \\ M_{12} \end{Bmatrix} = \frac{Eh^3}{12(1-\nu^2)} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \begin{Bmatrix} \kappa_{11} \\ \kappa_{22} \\ 2\kappa_{12} \end{Bmatrix} \tag{14}$$

Clearly extension and bending are decoupled for this particular case.

Eqs. (11), (12), and (13) form a system of 15 equations underpinning the Kirchhoff-Love model to be solved along with appropriate boundary conditions for 15 unknowns (three displacements,

six strain variables, and six stress resultants). Kinematics and kinetics remain the same no matter whether the structure is made of metals or composites and these equations have been implemented in many FEA codes which have plate/shell elements. Only difference is that the plate/shell stiffness matrix in Eq. (13) could be fully populated if the plate/shell is made of composites.

Although the Kirchhoff-Love model was originally developed based on a set of ad hoc assumptions as aforementioned, such assumptions are not used in MSG to derive this model. Also, although we used the familiar terms of A, B, D matrices describing the plate stiffness matrix as those used in the classical lamination theory (CLT), none of the assumptions associated with CLT is necessary for MSG to derive the Kirchhoff-Love model. Thus, the Kirchhoff-Love model here only refers to the model which has 15 field variables of x_1, x_2 governed by the 15 equations in Eqs. (11), (12) and (13). In other words, according to the MSG-based Kirchhoff-Love model, the transverse normal line could be deformed, not necessarily perpendicular to the reference surface, and all six stress components including both in-plane stresses and transverse stresses could exist.

4.3 Reissner-Mindlin Plate/Shell Model

When the thickness of the panel is not very small with respect to the in-plane dimensions, the Kirchhoff-Love model is inadequate and a refined model is needed. The next refinement is the so-called Reissner-Mindlin model due to independent contributions of Reissner and Mindlin to its development. The kinematics of the Reissner-Mindlin model contains five displacement variables including three displacements (u_1, u_2, u_3) and two rotations (θ_1, θ_2), and eight strain variables including in-plane strains $(\epsilon_{11}, \epsilon_{22}, \epsilon_{12})$, curvatures $(\kappa_{11}, \kappa_{22}, \kappa_{12})$, and transverse shear strains $(\gamma_{12}, \gamma_{13})$. For a plate, the strain-displacement relations are

$$\begin{aligned}
&\epsilon_{11} = \frac{\partial u_1}{\partial x_1}, \qquad \epsilon_{22} = \frac{\partial u_2}{\partial x_2}, \qquad 2\epsilon_{12} = \frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1} \\
&\kappa_{11} = \frac{\partial \theta_2}{\partial x_1}, \qquad \kappa_{22} = -\frac{\partial \theta_1}{\partial x_2}, \qquad \kappa_{12} = -\frac{\partial \theta_2}{\partial x_2} - \frac{\partial \theta_1}{\partial x_1} \\
&\gamma_{13} = \frac{\partial u_3}{\partial x_1} + \theta_2, \qquad \gamma_{23} = \frac{\partial u_3}{\partial x_2} - \theta_1
\end{aligned} \tag{15}$$

The kinetics of the Reissner-Mindlin model contains eight stress resultants ($N_{11}, N_{22}, N_{12}, M_{11}$, $M_{22}, M_{12}, N_{13}, N_{23}$) with N_{11}, N_{22}, N_{12} denoting in-plane forces, M_{11}, M_{22}, M_{12} denoting moments, and N_{13}, N_{23} denoting transverse shear forces. These kinetic variables are governed by the following

five equations of equilibrium:

$$\begin{aligned}
&\frac{\partial N_{11}}{\partial x_1}+\frac{\partial N_{12}}{\partial x_2}+p_1=0\\
&\frac{\partial N_{21}}{\partial x_1}+\frac{\partial N_{22}}{\partial x_2}+p_2=0\\
&\frac{\partial N_{13}}{\partial x_1}+\frac{\partial N_{23}}{\partial x_2}+p_3=0\\
&\frac{\partial M_{12}}{\partial x_1}+\frac{\partial M_{22}}{\partial x_2}-q_1-N_{23}=0\\
&\frac{\partial M_{11}}{\partial x_1}+\frac{\partial M_{21}}{\partial x_2}+q_2-N_{13}=0
\end{aligned} \tag{16}$$

The constitutive relations of the Reissner-Mindlin model can be expressed using the following matrix equation:

$$\begin{Bmatrix} N_{11}\\ N_{22}\\ N_{12}\\ M_{11}\\ M_{22}\\ M_{12}\\ N_{13}\\ N_{23} \end{Bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{16} & B_{11} & B_{12} & B_{16} & Y_{11} & Y_{12}\\ A_{12} & A_{22} & A_{26} & B_{12} & B_{22} & B_{26} & Y_{21} & Y_{22}\\ A_{16} & A_{26} & A_{66} & B_{16} & B_{26} & B_{66} & Y_{31} & Y_{32}\\ B_{11} & B_{12} & B_{16} & D_{11} & D_{12} & D_{16} & Y_{41} & Y_{42}\\ B_{12} & B_{22} & B_{26} & D_{12} & D_{22} & D_{26} & Y_{51} & Y_{52}\\ B_{16} & B_{26} & B_{66} & D_{16} & D_{26} & D_{66} & Y_{61} & Y_{62}\\ Y_{11} & Y_{21} & Y_{31} & Y_{41} & Y_{51} & Y_{61} & C_{11} & C_{12}\\ Y_{12} & Y_{22} & Y_{32} & Y_{42} & Y52 & Y_{62} & C_{12} & C_{22} \end{bmatrix} \begin{Bmatrix} \epsilon_{11}\\ \epsilon_{22}\\ 2\epsilon_{12}\\ \kappa_{11}\\ \kappa_{22}\\ 2\kappa_{12}\\ \gamma_{13}\\ \gamma_{23} \end{Bmatrix} \tag{17}$$

This 8×8 matrix is called the plate stiffness matrix and its inverse is called the plate compliance matrix for the Reissner-Mindlin model. $G_{ij}(i=1,2;j=1,2)$ denote the transverse shear stiffness terms and Y_{ij} $(i=1,\ldots,6;j=1,2)$ denote the coupling stiffness terms relating the classical plate deformation modes and transverse shear deformation modes. It is noted that A,B,D matrices could be different from those in Eq. (13) due to possible nonzero Y_{ij} $(i=1,\ldots,6;j=1,2)$.

Eqs. (15), (16), and (17) form a system of 21 equations underpinning the Reissner-Mindlin model to be solved along with appropriate boundary conditions for 21 unknowns (five displacement variables, eight strain variables, and eight stress resultants). Kinematics and kinetics remain the same no matter whether the structure is made of metals or composites and these equations have been implemented in many FEA codes which have plate/shell elements. Only difference is that the plate/shell stiffness matrix in Eq. (17) could be fully populated if the plate/shell is made of composites.

Although the Reissner-Mindlin model was originally developed based on a set of ad hoc assumptions, such assumptions are not used in MSG to derive this model. Thus, the Reissner-Mindlin model here only refers to the model which has 21 field variables of x_1, x_2 governed by the 21 equations in Eqs. (15), (16) and (17). The deformation and stress state of the structure are not assumed a priori but determined by MSG.

4.4 Euler-Bernoulli Beam Model

The kinematics of the Euler-Bernoulli beam model contains four displacement variables u_1, u_2, u_3, θ_1 with u_1, u_2, u_3 describing displacements in three directions and θ_1 describing the twist angle, and four strain variables including axial strain γ_{11}, twist rate κ_{11}, and curvatures κ_{12}, κ_{13} around x_2 and x_3, respectively. The strain-displacement relations are

$$\gamma_{11} = \frac{\mathrm{d}u_1}{\mathrm{d}x_1}, \qquad \kappa_{11} = \frac{\mathrm{d}\theta_1}{\mathrm{d}x_1}, \qquad \kappa_{12} = -\frac{\mathrm{d}^2 u_3}{\mathrm{d}x_1^2}, \qquad \kappa_{13} = \frac{\mathrm{d}^2 u_2}{\mathrm{d}x_1^2} \tag{18}$$

The kinetics of the Euler-Bernoulli beam model contains four stress resultants (F_1, M_1, M_2, M_3) with F_1 denoting axial force and M_1, M_2, M_3 denoting moments about three directions. These kinetic variables are governed by the following four equations of equilibrium:

$$\begin{aligned}
&\frac{\mathrm{d}F_1}{\mathrm{d}x_1} + p_1 = 0 \\
&\frac{\mathrm{d}M_1}{\mathrm{d}x_1} + q_1 = 0 \\
&\frac{d^2 M_2}{\mathrm{d}x_1^2} + p_3 + \frac{\mathrm{d}q_2}{\mathrm{d}x_1} = 0 \\
&\frac{d^2 M_3}{\mathrm{d}x_1^2} - p_2 + \frac{\mathrm{d}q_3}{\mathrm{d}x_1} = 0
\end{aligned} \tag{19}$$

where p_1, p_2, p_3 are equivalent forces and q_1, q_2, q_3 are equivalent moments in three directions, distributed along the reference line.

The constitutive relations of the Euler-Bernoulli beam model can be expressed using the following matrix equation:

$$\begin{Bmatrix} F_1 \\ M_1 \\ M_2 \\ M_3 \end{Bmatrix} = \begin{bmatrix} C_{11}^b & C_{12}^b & C_{13}^b & C_{14}^b \\ C_{12}^b & C_{22}^b & C_{23}^b & C_{24}^b \\ C_{13}^b & C_{23}^b & C_{33}^b & C_{34}^b \\ C_{14}^b & C_{24}^b & C_{34}^b & C_{44}^b \end{bmatrix} \begin{Bmatrix} \gamma_{11} \\ \kappa_{11} \\ \kappa_{12} \\ \kappa_{13} \end{Bmatrix} \tag{20}$$

This 4×4 matrix is commonly called the beam stiffness matrix and its inverse is called the beam compliance matrix for the Euler-Bernoulli model. If a beam is made of a single isotropic material, and the origin of the cross-sectional coordinates x_2, x_3 is chosen to be at the tension center of the cross-section and x_2, x_3 are chosen to align with the principal bending directions of the beam, the constitutive relations can be written as

$$\begin{Bmatrix} F_1 \\ M_1 \\ M_2 \\ M_3 \end{Bmatrix} = \begin{bmatrix} EA & 0 & 0 & 0 \\ 0 & GJ & 0 & 0 \\ 0 & 0 & EI_2 & 0 \\ 0 & 0 & 0 & EI_3 \end{bmatrix} \begin{Bmatrix} \gamma_{11} \\ \kappa_{11} \\ \kappa_{12} \\ \kappa_{13} \end{Bmatrix} \tag{21}$$

Clearly the beam stiffness matrix becomes a diagonal matrix for this particular case and the diagonal terms are the well known engineering constants including extension stiffness EA, torsional

stiffness GJ, and bending stiffnesses EI_2 and EI_3 for bending about x_2 and x_3 respectively.

Eqs. (18), (19), and (20) form a system of 12 equations underpinning the Euler-Bernoulli model to be solved along with appropriate boundary conditions for 12 unknowns (four displacement variables, four strain variables, and four stress resultants). Kinematics and kinetics remain the same no matter whether the structure is made of metals or composites and these equations have been implemented in many FEA codes which have beam elements. Only difference is that the beam stiffness matrix in Eq. (20) could be fully populated if the beam is made of composites. It is noted that many beam problems, particularly those with a uniform cross-section, can be solved analytical using what we have learned in undergraduate mechanics of materials.

Although the Euler-Bernoulli model was originally developed based on a set of ad hoc assumptions including the cross-section being rigid in plane, perpendicular to the reference line, and uniaxial stress assumption. However, such assumptions are not used in MSG to derive this model. Thus, the Euler-Bernoulli model here only refers to the model which has 12 field variables of x_1 governed by the 12 equations in Eqs. (18), (19) and (20). In other words, according to the MSG-based Euler-Bernoulli model, the cross-section could be deformed, not necessarily perpendicular to the reference line, and all six stress components could exist. Actually, the slender structure which is modeled using an MSG-based beam model may not even have clearly defined cross-sections. As long as an SG for the slender structure can be defined, an MSG-based beam model can be constructed for the structural analysis.

4.5 Timoshenko Beam Model

The kinematics of the Timoshenko beam model contains six displacement variables $(u_1, u_2, u_3, \theta_1, \theta_2, \theta_3)$ with u_1, u_2, u_3 describing displacements in three directions and $\theta_1, \theta_2, \theta_3$ describing rotations about three directions and six strain variables including axial strain γ_{11}, transverse shear strains $(\gamma_{12}, \gamma_{13})$, twist rate κ_{11}, and curvatures $(\kappa_{12}, \kappa_{13})$. The strain-displacement relations are

$$\begin{aligned}
&\gamma_{11} = \frac{\mathrm{d}u_1}{\mathrm{d}x_1}, \qquad \gamma_{12} = -\theta_3 + \frac{\mathrm{d}u_2}{\mathrm{d}x_1}, \qquad \gamma_{13} = \theta_2 + \frac{\mathrm{d}u_3}{\mathrm{d}x_1} \\
&\kappa_{11} = \frac{\mathrm{d}\theta_1}{\mathrm{d}x_1}, \qquad \kappa_{12} = \frac{\mathrm{d}\theta_2}{\mathrm{d}x_1}, \qquad \kappa_{13} = \frac{\mathrm{d}\theta_3}{\mathrm{d}x_1}
\end{aligned} \tag{22}$$

The kinetics of the Timoshenko beam model contains six stress resultants $(F_1, F_2, F_3, M_1, M_2, M_3)$ with F_1 denoting axial force, F_2 and F_3 denoting transverse shear forces, and M_1, M_2, M_3 denoting bending moments about three directions. These kinetic variables are functions of x_1 only and they

are governed by the following six equations of equilibrium:

$$\begin{aligned}
&\frac{\mathrm{d}F_1}{\mathrm{d}x_1} + p_1 = 0\\
&\frac{\mathrm{d}F_2}{\mathrm{d}x_1} + p_2 = 0\\
&\frac{\mathrm{d}F_3}{\mathrm{d}x_1} + p_3 = 0\\
&\frac{\mathrm{d}M_1}{\mathrm{d}x_1} + q_1 = 0\\
&\frac{dM_2}{\mathrm{d}x_1} - F_3 + q_2 = 0\\
&\frac{dM_3}{\mathrm{d}x_1} + F_2 + q_3 = 0
\end{aligned} \tag{23}$$

The constitutive relations of the Timoshenko beam model can be expressed using the following matrix equation:

$$\begin{Bmatrix} F_1\\ F_2\\ F_3\\ M_1\\ M_2\\ M_3 \end{Bmatrix} = \begin{bmatrix} C_{11}^b & C_{12}^b & C_{13}^b & C_{14}^b & C_{15}^b & C_{16}^b\\ C_{12}^b & C_{22}^b & C_{23}^b & C_{24}^b & C_{25}^b & C_{26}^b\\ C_{13}^b & C_{23}^b & C_{33}^b & C_{34}^b & C_{35}^b & C_{36}^b\\ C_{14}^b & C_{24}^b & C_{34}^b & C_{44}^b & C_{45}^b & C_{46}^b\\ C_{15}^b & C_{25}^b & C_{35}^b & C_{45}^b & C_{55}^b & C_{56}^b\\ C_{16}^b & C_{26}^b & C_{36}^b & C_{46}^b & C_{56}^b & C_{66}^b \end{bmatrix} \begin{Bmatrix} \gamma_{11}\\ \gamma_{12}\\ \gamma_{13}\\ \kappa_{11}\\ \kappa_{12}\\ \kappa_{13} \end{Bmatrix} \tag{24}$$

This 6×6 matrix is called the beam stiffness matrix and its inverse is called the beam compliance matrix for the Timoshenko model. It is noted that C_{ij}^b $(i = 1, 2, 3, 4; j = 1, 2, 3, 4)$ in this stiffness matrix could be different from those in Eq. (20).

Eqs. (22), (23), and (24) form a system of 18 equations to be solved along with appropriate boundary conditions for 18 unknowns (six displacement variables, six strain variables, and six stress resultants). Kinematics and kinetics remain the same no matter whether the structure is made of metals or composites and these equations have been implemented in many FEA codes which have beam elements. Only difference is that the beam stiffness matrix in Eq. (24) could be fully populated if the beam is made of composites.

It is noted that although the Timoshenko model was originally developed based on a set of ad hoc assumptions including the cross-section being rigid in plane, remaining plane during deformation, and uniaxial stress assumption. However, such assumptions are not used in MSG to derive this model. Thus, the Timoshenko model here only refers to the model which has 18 field variables of x_1 governed by the 18 equations in Eqs. (22), (23) and (24).

5 Mechanics of Structure Genome (MSG)

5.1 The Concept of Structure Gene (SG) and Structure Genome

Inspired by the fundamental role of the gene for an organism's growth and development, we extrapolate this word into non-biological contexts to define SG as the *smallest mathematical building block* of a structure. SG contains all the constitutive information needed for a structure in the same fashion as the gene contains all the genetic information for an organism. A structure may have more than one fundamental building blocks. The complete set of structure genes present in a structure is called Structure Genome for that structure. It is noted that if one is using SwiftComp^{TM} for multiphysics behavior other than structural behavior, an SG should be interpreted as the smallest mathematical building block (or gene) of the material instead.

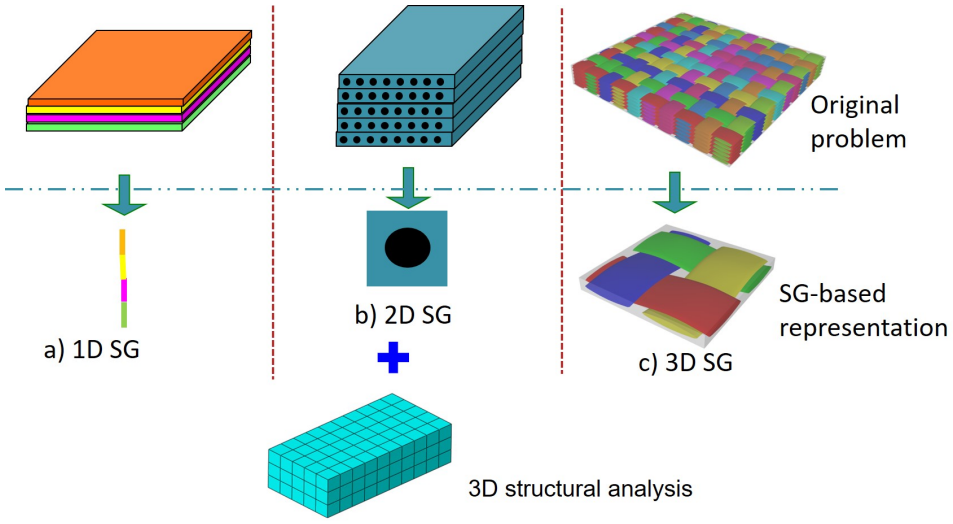

Figure 3: Analysis of 3D heterogeneous structures approximated by a constitutive modeling over an SG and a corresponding 3D macroscopic structural analysis.

5.2 SG for 3D Structures

As shown in Figure 3, analyses of 3D heterogeneous structures can be approximated by a 3D macroscopic structural analysis with the material properties provided by a constitutive modeling of an SG. For 3D structures, SG serves a similar role as the representative volume element (RVE) in micromechanics. However, they are significantly different so that the new term, SG, is used to avoid confusion. For example, for a structure made of composites featuring 1D heterogeneity (e.g. composite laminates made of layers with different orientations, Figure 3a), the SG will be the transverse normal line with segments denoting the corresponding layers. One can mathematically repeat this line in-plane to build the composite laminate. One possible application of 1D SG for 3D structures is to compute the effective 3D properties of a composite laminate. The constitutive modeling over the 1D SG can compute the complete set of 3D properties and local fields. Such

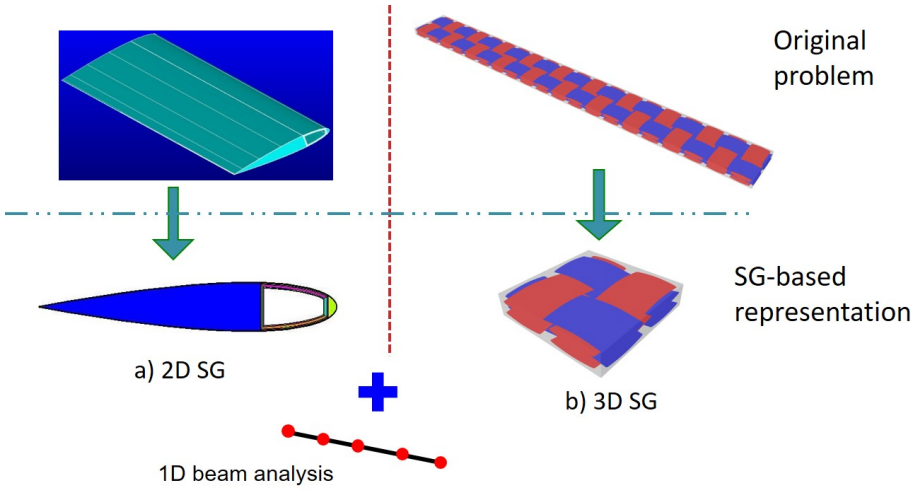

Figure 4: Analysis of beam-like structures approximated by a constitutive modeling over an SG and a corresponding 1D beam analysis.

applications of SG are not equivalent to RVE. For a structure made of composites featuring 2D heterogeneity (e.g. continuous unidirectional fiber reinforced composites, Figure 3b), the SG will be 2D. Although 2D RVEs are also used in micromechanics, only in-plane properties and in-plane local fields can be obtained from common RVE-based models. If the complete set of properties are needed for the 3D structural analysis, a 3D RVE is usually required [7], while a 2D domain is sufficient if it is modeled using SG-based models (Figure 3b). For a structure made of composites featuring 3D heterogeneity (e.g. textile composites, Figure 3c), the SG will be a 3D volume. Although a 3D SG for 3D structures represents the most similar case to RVE, boundary conditions in terms of displacements and tractions indispensable in RVE-based models are not needed for SG-based models.

5.3 SG for Dimensionally Reducible Structures

SG also allows direct connection of microstructure with the beam/plate/shell analyses. For example, the structural analysis of slender (beam-like) structures can use beam elements (Figure 4). If the beam has uniform cross-sections which could be made of isotropic homogeneous materials or anisotropic heterogeneous materials (Figure 4a), its SG is the 2D cross-sectional domain because the cross-section can be projected along the beam reference line to form the beam-like structure. This inspires a *new perspective* toward beam modeling, a traditional branch of structural mechanics. If the beam reference line is considered as a general 1D continuum, every material point of this continuum has a cross-section as its microstructure. In other words, *constitutive modeling for beams can be effectively viewed as an application of micromechanics.* If the beam is also heterogeneous in the spanwise direction (Figure 4b), a 3D SG is needed to describe the microstructure of the 1D continuum, the behavior of which is governed by the 1D beam analysis. The constitutive modeling over an SG should compute the beam stiffness for the beam analysis, and also the

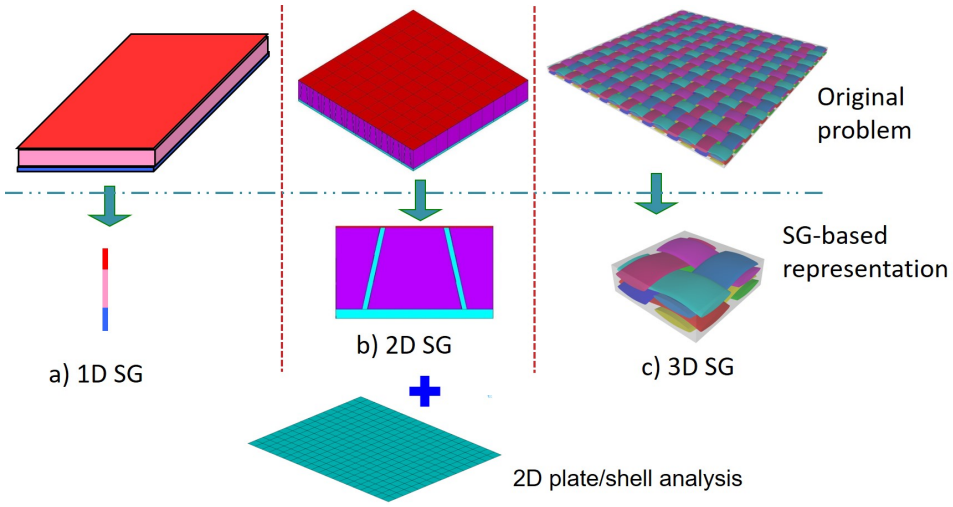

Figure 5: Analysis of plate-like structures approximated by a constitutive modeling over SG and a corresponding 2D plate analysis.

complete 3D displacement, stress, and strain fields within the original structure. The concept of SG provides a unified treatment of structural modeling and micromechanics modeling and enables us to collapse the cross-section or a 3D beam segment into a material point for a beam analysis over the reference line with a possible, fully populated stiffness matrix simultaneously accounting for all the deformation modes of a certain model such as the Euler-Bernoulli beam model (extension, torsion, and bending in two directions) or the Timoshenko beam model (extension, torsion, bending in two directions, and shear in two directions). The beam model constructed this way can easily handle buildup structures as long as their external contours look like a beam such as high aspect ratio wings, or as long as the analyst wants to model a slender structure using a beam element.

If the structural analysis uses plate/shell elements, SG can also be chosen properly. For illustrative purpose, typical SGs of plate-like structures are sketched in Figure 5. If the plate-like structures feature no in-plane heterogeneities (in other words, the structure is a laminate made of homogeneous layers) (Figure 5a), the SG is the transverse normal line with each segment denoting the corresponding layer. For a sandwich panel with a core corrugated in one direction (Figure 5b), the SG is 2D. If the panel is heterogeneous in both in-plane directions (Figure 5c), such as a stiffened panel with stiffeners running in both directions or a panel made of 3D textile composites, the SG is 3D. Despite the different dimensionalities of the SGs, the constitutive modeling should compute structural properties for the corresponding structural analysis (such as the A, B, and D matrices for the Kirchhoff-Love plate model) and relations to express the original 3D fields in terms of the global behavior (e.g. moments, curvatures, etc.) obtained from the plate/shell analysis. It is known that theories of plates/shells traditionally belong to structural mechanics, but the constitutive modeling of these structures can be treated as special micromechanics applications using the SG concept. For a plate/shell-like structure, if the reference surface is considered as a general 2D continuum, every material point of this continuum has an associated SG as its microstructure. Plate/shell models constructed using the SG concept can handle buildup structures as long as their

external contours look like plates or shells or we want to model a structure using plate or shell elements.

It is easy to identify SGs for periodic structures as shown in Figures 3, 4, and 5. For structures which are not globally periodic, we usually assume that the structure is at least periodic in the neighborhood of a material point in the macroscopic structural analysis, the so-called local periodicity assumption implicit in all multiscale modeling approaches. For nonlinear behavior, it is also possible that the smallest mathematical building block of the structure is not sufficient as the characteristic length scale of the nonlinear behavior may cover several building blocks. For this case, the SG should be interpreted as the smallest mathematical building block necessary to represent the nonlinear behavior. In the most general cases, SG can be considered as the block of material corresponding to the element in finite element mesh of the macroscopic structural analysis. In the extreme case, a structure could have as many SGs as the number of elements in the macroscopic structural analysis. To be rigorous, such extreme cases should be handled by DNS of all the details in the macroscopic structural analysis because the local periodicity assumption is not valid. However, in real structural design and analysis, the multiscale approach is still practiced. For example, for a wind turbine blade, the cross section is changing significantly along the span. In preliminary design of wind turbine blades, the blade is cut into hundreds of stations with each cross-section considered as an SG for a corresponding beam element. Such a modeling approach is usually needed for the multibody dynamic analysis at the system level. According to the taxonomy in biology, genome is defined as the complete set of genes present in an organism. In this sense, we can consider the totality of all the cross-sections we used in this analysis as the *structure genome* of the wind turbine blade.

SG serves as the link between the original structure with microscopic details and the macroscopic structural analysis. Here, the terms "microstructure" and "microscopic details" are used in a general sense: any details explicitly existing in an SG but not in the macroscopic structural analysis are termed microscopic details. The real structure with microscopic details is termed as the original structure and the structure used in the macroscopic structural analysis is termed as the macroscopic structural model. It is also interesting to point out the relation between the SG concept and the idea of sub-structuring or super-element, which is commonly used in engineering. A line element in the global analysis could correspond to a box beam made of four laminated walls, and a surface element could correspond to a sandwich panel with laminated face sheets and a corrugated core. For these cases, MSG provides a rigorous and systematic approach to compute the constitutive models for the line and surface elements and the local fields (displacements, stresses, and strains) within the original structures.

For some applications, the local periodicity assumption may not be valid. In other words, SG cannot be considered as a point in the macroscopic analysis. To remove this assumption, SwiftComp$^{\text{TM}}$ 2.1 added the capability for performing constitutive modeling of a SG meshed with many 3D elements to be a single, homogeneous 3D 8-node or 20-node element. SwiftComp$^{\text{TM}}$ will perform homogenization of the SG to obtain the effective element stiffness matrix for the homogeneous element, which can used as input for the macroscopic structural analysis using the homogeneous element. The macroscopic analysis will compute the nodal values of the homogeneous

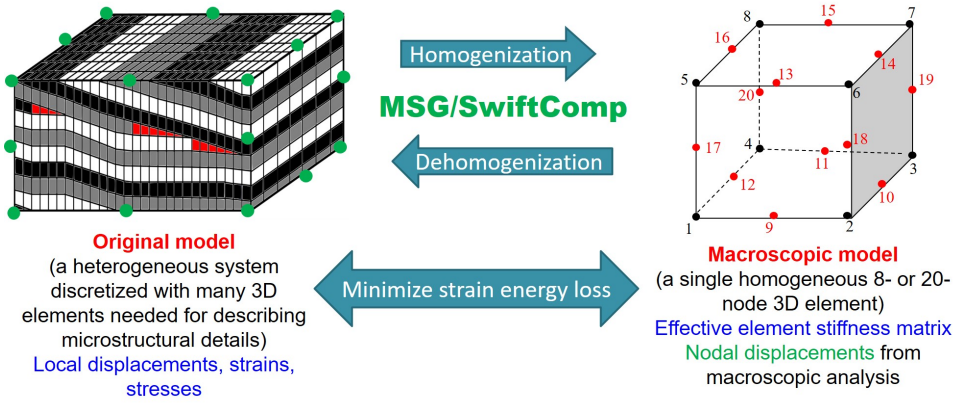

Figure 6: The basic idea of constitutive modeling of a block of 3D elements to be a single 3D element.

element which can be used as input for SwiftComp^{TM} to perform dehomogenization to compute the 3D fields (displacements/stresses/strains) within the SG. It is noted that the macroscopic structural analysis using homogeneous elements can be handled by FEA codes which allow users to define their own element type such as the Abaqus UEL subroutine. The basic idea is illustrated in Figure 6.

5.4 MSG-based Multiscale Structural Modeling

The multiscale structural modeling approach based on MSG can be used to fill the gap between materials genome and macroscopic structural analysis and can directly connect with simple structural elements (beam/plate/shell/3D solid elements) available in standard FEA software packages (see Figure 2).

As shown in Figure 7, MSG starts from the original model formulated in terms of 3D continuum mechanics. We first identify SG for a structure, then use the principle of minimum information loss (PMIL) to decouple the original problem to a constitutive modeling over the SG and a structural analysis. The constitutive modeling has been implemented in SwiftComp^{TM}for computing the effective properties needed in the structural analysis and local fields within the original structure. The structural analysis can be formulated as a geometrically exact continuum theory and all the approximations are confined in the constitutive modeling, accuracy of which is guaranteed to be the best by PMIL.

As the macroscopic structural analysis can be easily handled by standard FEA software packages, this MSG-based multiscale structural modeling approach enables these FEA software packages to model composites as a *black aluminum* in the macroscopic structural analysis with minimized loss of accuracy. But this "black aluminum" is a 1D (beam), 2D (plate/shell), or 3D (solid) continuum featuring general anisotropic constitutive relations, not an isotropic material as traditionally implied by this commonly used term. The unique features of this multiscale modeling approach

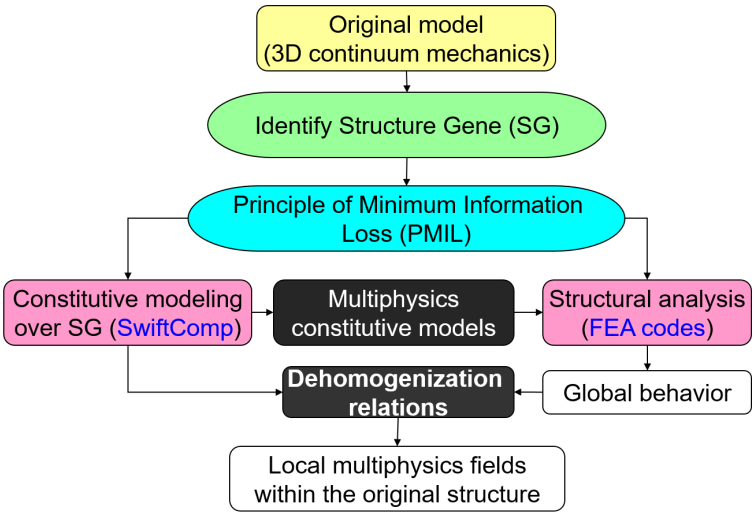

Figure 7: Work flow of MSG-based multiscale modeling.

based on MSG and its companion code SwiftComp$^{\text{TM}}$ are:

- Use SG to fill the gap between materials genome and the macroscopic structural analysis. Intellectually, SG enables us to view constitutive modeling of structures as an application of micromechanics. Technically, SG empowers us to systematically model complex buildup structures with heterogeneities of length scales comparable to the smallest structural dimension.

- Use PMIL based on the powerful variational asymptotic method (VAM) to avoid apriori assumptions commonly invoked in other approaches, providing the most mathematical rigor and the best engineering generality.

- Decouple the original problem into two sets of analyses: a constitutive modeling and a structural analysis. This allows the structural analysis to be formulated exactly as a general (1D, 2D, or 3D) continuum, the analysis of which is readily available in commercial FEA software packages and confines all approximations to the constitutive modeling, the accuracy of which is guaranteed to be the best by PMIL.

- Maintain the engineering simplicity and legacy by repacking the refined asymptotically correct functionals into common engineering models so that models constructed using this approach can be incorporated into standard FEA software packages.

Since most of the theoretical details of SG are presented in its relevant publications [1, 2, 3, 4, 8, 9, 10, 11, 12, 13, 14, 15, 16], this manual will only serve to help readers get started using SwiftComp$^{\text{TM}}$ to solve their own constitutive modeling problems. In the following, we will address its conventions, inputs, outputs, maintenance, and tech support.

Figure 8: 1D element nodal numbering.

6 SwiftComp^{TM} Conventions

To understand the inputs and interpret outputs of the program correctly, we need to explain some conventions used in SwiftComp^{TM}.

6.1 Elements

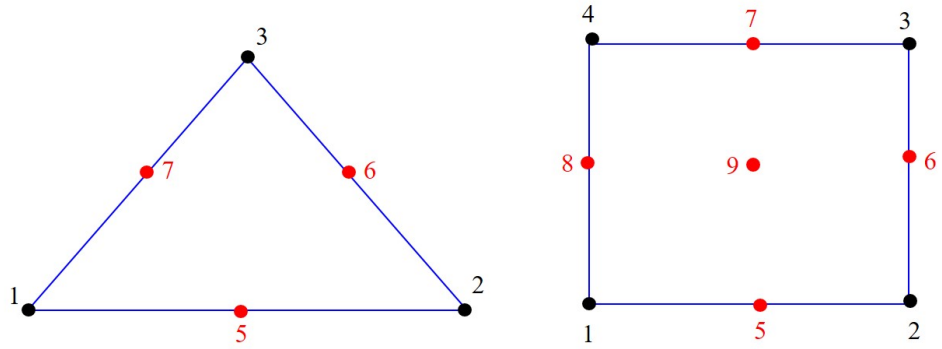

Figure 9: 2D element nodal numbering.

SwiftComp^{TM} meshes 1D SGs using two-node, three-node, four-node, or five-node elements for as shown in Figure 8. Nodes 3, 4, 5 are optional and one or more of these nodes can be missing for a valid 1D element. It is recommended to use 2-node elements for 3D structures with a 1D SG (see Figure 3a) and 5-node elements for 2D plate/shell models with a 1D SG (see Figure 5a).

SwiftComp^{TM} meshes 2D SGs using either triangular or quadrilateral elements as shown in Figure 9. It is also shown in the figure that SwiftComp^{TM} numbers the nodes of each 2D elements in the counterclockwise direction. Nodes 1, 2, and 3 of the triangular elements and nodes 1, 2, 3, and 4 of the quadrilateral elements are at the corners. For triangular element, the fourth node is zero to inform SwiftComp^{TM} that it is a triangular element. Nodes 5, 6, 7 of the triangular elements and nodes 5, 6, 7, 8, 9 of quadrilateral elements are optional. Any one or more of these nodes can be missing for a valid 2D element.

SwiftComp^{TM} meshes 3D SGs using tetrahedral elements, brick elements, or wedge elements as shown in Figure 10. For tetrahedral elements, the fifth node is zero to inform SwiftComp^{TM} that

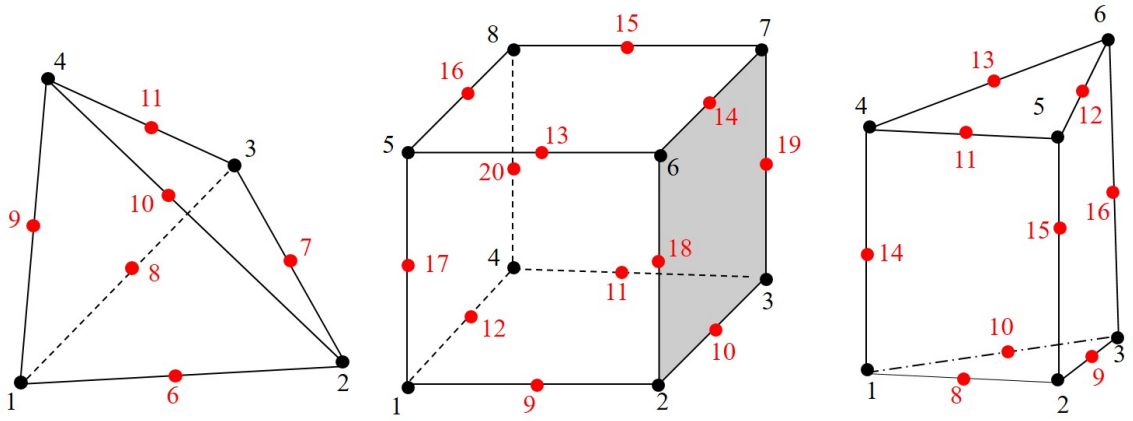

Figure 10: 3D element nodal numbering.

it is a tetrahedral element. For wedge elements, the seventh node is zero to inform SwiftComp$^{\text{TM}}$ that it is a wedge element. The nodes other than the corners are optional. Any one or more of these nodes can be missing for a valid 3D element.

6.2 Local Coordinate System, Elemental Coordinate System, and Material Coordinate System

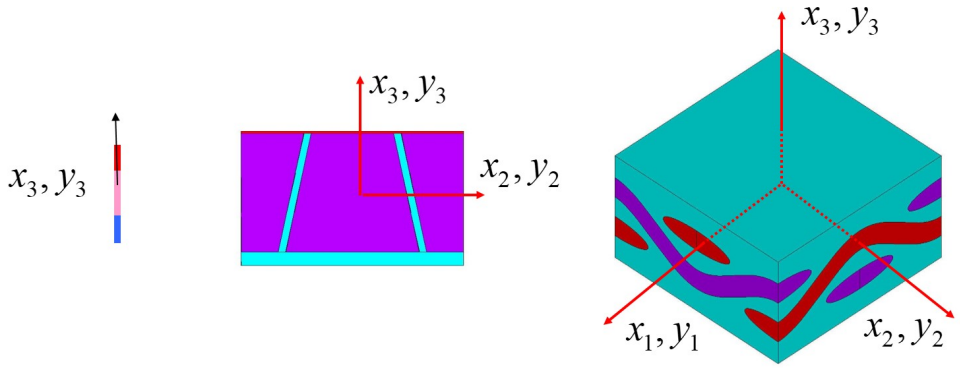

Figure 11: Local coordinate system describing SG.

First, SwiftComp$^{\text{TM}}$ uses a right-hand Cartesian coordinate system, also called the *local coordinate system*, denoted as y_1, y_2 and y_3, to describe a 3D SG, y_2 and y_3 to describe a 2D SG, and y_3 to describe a 1D SG (see Figure 11). y_1, y_2, y_3 are parallel to the *global coordinates* x_1, x_2, x_3, respectively. The global coordinates x_1, x_2, x_3 are used to describe the original structure and the macroscopic structure. Note that if the material properties are provided in a coordinate system different from y_1, y_2, y_3, an additional coordinate system called the *material coordinate system* should be defined and a transformation of the material properties from the material coordinate system

into those expressed in the local coordinate system is automatically carried out by SwiftComp$^{\text{TM}}$.

In SwiftComp$^{\text{TM}}$, an elemental coordinate system y_i' can be defined for each element denoted by three points a, b, c, with the line from point c to point a denoting y_1' direction and the line from point c to point b denoting a line located in the $y_1' - y_2'$ plane; see Figure 12 for a sketch. Speaking in the language of vectors, the new coordinate system is defined by three points with position vectors in the local coordinate system (y_i with $\hat{\boldsymbol{e}}_i$ as the unit vectors) by $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$. $\boldsymbol{a} - \boldsymbol{c}$ denotes $\hat{\boldsymbol{e}}_1'$, $\boldsymbol{b} - \boldsymbol{c}$ is a vector in the $y_1' - y_2'$ plane. With this information, one can compute the direction cosine matrix relating y_i to y_i' according to the following steps:

- Obtain $\hat{\boldsymbol{e}}_1'$ through normalization of $\boldsymbol{a} - \boldsymbol{c}$: $\hat{\boldsymbol{e}}_1' = \frac{\boldsymbol{a}-\boldsymbol{c}}{|\boldsymbol{a}-\boldsymbol{c}|}$;
- Obtain $\hat{\boldsymbol{e}}_3'$ through normalization of the cross product of $\hat{\boldsymbol{e}}_1'$ and $\boldsymbol{b} - \boldsymbol{c}$: $\hat{\boldsymbol{e}}_3' = \frac{\hat{\boldsymbol{e}}_1' \times (\boldsymbol{b}-\boldsymbol{c})}{|\hat{\boldsymbol{e}}_1' \times (\boldsymbol{b}-\boldsymbol{c})|}$;
- Obtain $\hat{\boldsymbol{e}}_2'$ through the cross product of $\hat{\boldsymbol{e}}_3'$ and $\hat{\boldsymbol{e}}_1'$: $\hat{\boldsymbol{e}}_2' = \hat{\boldsymbol{e}}_3' \times \hat{\boldsymbol{e}}_1'$.

SwiftComp$^{\text{TM}}$ allows the user to define the material properties in the local coordinate system y_i or in the material coordinate system. The material coordinate system could be the elemental coordinate system or a coordinate system defined in such a way that it can be obtained by a simple rotation about y_3' of the elemental coordinate system. Clearly for composite laminates, this simple rotation corresponds to the layup angle.

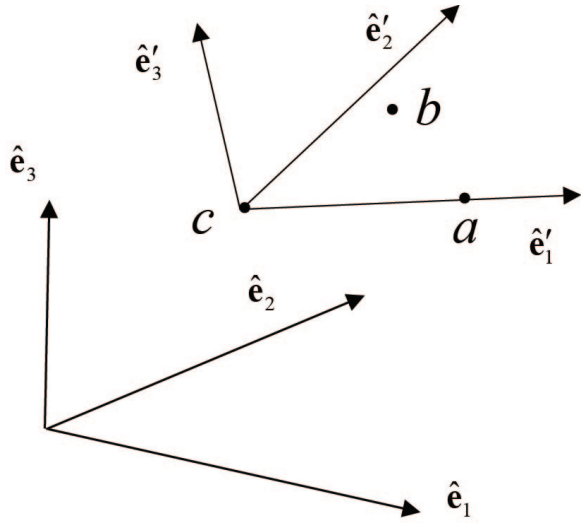

Figure 12: Elemental coordinate system defined by three points.

6.3 Constituent Constitutive Models

Generally speaking, the constituents contained in a SG could be responsive to thermal, mechanical, electric, and magnetic fields. If these effects are not coupled, the linear elastic behavior can be

modeled using the Hooke's law in Eq. (3).

To deal with uncoupled thermal, electric, and magnetic effects, conduction can be modeled using the following constitutive relation:

$$\begin{Bmatrix} q_1 \\ q_2 \\ q_3 \end{Bmatrix} = - \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{12} & k_{22} & k_{23} \\ k_{13} & k_{23} & k_{33} \end{bmatrix} \begin{Bmatrix} T_{,1} \\ T_{,2} \\ T_{,3} \end{Bmatrix} \tag{25}$$

where q_i is the heat flux, k_{ij} is the conductivity, and $T_{,i}$ is the gradient of the temperature T. Since conduction is mathematically analogous to electrostatics, magnetostatics, and diffusion, SwiftComp$^{\text{TM}}$ can also be used to predict effective dielectric, magnetic, and diffusive properties of composite materials and the corresponding local fields. For example, to obtain the effective dielectric properties, we just need to let q_i denote the electric displacements, T denote the electric potential, and k_{ij} denote the corresponding dielectric properties.

For coupled mutliphysics modeling, we will have piezoelectric and piezomagnetic effects as well as pyroelectric, pyromagnetic, and electromagnetic effects. For linear behavior among all these fields, the constitutive equations can be expressed as:

$$\begin{aligned} \sigma_{ij} &= C_{ijkl}\varepsilon_{kl} - e_{kij}E_k - q_{kij}H_k + \Lambda_{ij}\theta \\ D_i &= e_{ikl}\varepsilon_{kl} + k_{ik}E_k + a_{ik}H_k + p_i\theta \\ B_i &= q_{ikl}\varepsilon_{kl} + a_{ik}E_k + \mu_{ik}H_k + m_i\theta \end{aligned} \tag{26}$$

where C_{ijkl}, e_{kij}, q_{kij}, and Λ_{ij} are the elastic, the piezoelectric, the piezomagnetic, and the thermal stress tensors, respectively (note that $\Lambda_{ij} = -C_{ijkl}\alpha_{kl}$ with α_{kl} as the thermal expansion tensor); σ_{ij} and ε_{ij} are the stress tensor and strain tensor, respectively; k_{ik}, a_{ik}, and μ_{ik} are the dielectric, electromagnetic, and magnetic permeability tensors, respectively; and p_i and m_i are the pyroelectric and pyromagnetic vectors, and D_i, E_k, B_i, and H_k are the electric displacement, electric field, magnetic induction, and magnetic field vectors, respectively. θ denotes the difference between the actual temperature and the reference temperature. SwiftComp$^{\text{TM}}$ does not restrict θ to be small. If θ is not small, Λ_{ij}, p_i, m_i are not the tangent or instantaneous properties, but the secant properties which are defined as average over a change of temperature. For example, let $\alpha_t(T)$ denote the tangent or instantaneous coefficient of thermal expansion (CTE), the secant CTE is defined as

$$\alpha(T) = \frac{1}{T - T_1}\int_{T_1}^{T} \alpha_t(\zeta)d\zeta = \frac{1}{\theta}\int_{T_1}^{T_1+\theta} \alpha_t(\zeta)d\zeta \tag{27}$$

with T_1 as the reference temperature and $\theta = T - T_1$. For convenience, SwiftComp$^{\text{TM}}$ uses tangent or instantaneous properties for α_{ij}, p_i, m_i as inputs and computes the secant properties internally for constitutive modeling of temperature dependent properties.

Linear multiphysics behavior is modeled based on the following energy functional corresponding to the constitutive equation in Eq. (26):

$$U = \frac{1}{2}\epsilon^T L\epsilon + \epsilon^T\beta\theta - \int_{T_1}^{T}\int_{T_1}^{\zeta} \frac{c_v(0,\rho)}{\rho}d\rho d\zeta \tag{28}$$

where

$$\epsilon = \lfloor \varepsilon_{11} \;\; \varepsilon_{22} \;\; \varepsilon_{33} \;\; 2\varepsilon_{23} \;\; 2\varepsilon_{13} \;\; 2\varepsilon_{12} \;\; -E_1 \;\; -E_2 \;\; -E_3 \;\; -H_1 \;\; -H_2 \;\; -H_3 \rfloor^T \tag{29}$$

is a multiphysical field array containing the 3D strain field ε_{ij}, the 3D electric field E_i, and the 3D magnetic field H_i. The conjugate multiphysical field array σ can be expressed as

$$\sigma = \lfloor \sigma_{11} \;\; \sigma_{22} \;\; \sigma_{33} \;\; \sigma_{23} \;\; \sigma_{13} \;\; \sigma_{12} \;\; D_1 \;\; D_2 \;\; D_3 \;\; B_1 \;\; B_2 \;\; B_3 \rfloor^T \tag{30}$$

L is a 12×12 multiphysics matrix containing all the necessary material constants for characterizing fully coupled thermoelectromagnetoelastic materials such that

$$L = \begin{bmatrix} C & e & q \\ e^T & -k & -a \\ q^T & -a^T & -\mu \end{bmatrix} \tag{31}$$

where C is a 6×6 submatrix for elastic constants, e is a 6×3 submatrix for piezoelectric coefficients, q is a 6×3 submatrix for piezomagnetic coefficients, k is a 3×3 submatrix for dielectric coefficients, a is a 3×3 submatrix for electromagnetic coefficients, and μ is a 3×3 submatrix for magnetic permeability. Note C, k, μ, a are symmetric matrices. The explicit form of the 12×12 matrix is as follows

$$\begin{bmatrix}
C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} & e_{11} & e_{21} & e_{31} & q_{11} & q_{21} & q_{31} \\
C_{12} & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} & e_{12} & e_{22} & e_{32} & q_{12} & q_{22} & q_{32} \\
C_{13} & C_{23} & C_{33} & C_{34} & C_{35} & C_{36} & e_{13} & e_{23} & e_{33} & q_{13} & q_{23} & q_{33} \\
C_{14} & C_{24} & C_{34} & C_{44} & C_{45} & C_{46} & e_{14} & e_{24} & e_{34} & q_{14} & q_{24} & q_{34} \\
C_{15} & C_{25} & C_{35} & C_{45} & C_{55} & C_{56} & e_{15} & e_{25} & e_{35} & q_{15} & q_{25} & q_{35} \\
C_{16} & C_{26} & C_{36} & C_{46} & C_{56} & C_{66} & e_{16} & e_{26} & e_{36} & q_{16} & q_{26} & q_{36} \\
e_{11} & e_{12} & e_{13} & e_{14} & e_{15} & e_{16} & -k_{11} & -k_{12} & -k_{13} & -a_{11} & -a_{12} & -a_{13} \\
e_{21} & e_{22} & e_{23} & e_{24} & e_{25} & e_{26} & -k_{12} & -k_{22} & -k_{23} & -a_{12} & -a_{22} & -a_{23} \\
e_{31} & e_{32} & e_{33} & e_{34} & e_{35} & e_{36} & -k_{13} & -k_{23} & -k_{33} & -a_{13} & -a_{23} & -a_{33} \\
q_{11} & q_{12} & q_{13} & q_{14} & q_{15} & q_{16} & -a_{11} & -a_{21} & -a_{31} & -\mu_{11} & -\mu_{12} & -\mu_{13} \\
q_{21} & q_{22} & q_{23} & q_{24} & q_{25} & q_{26} & -a_{12} & -a_{22} & -a_{32} & -\mu_{12} & -\mu_{22} & -\mu_{23} \\
q_{31} & q_{32} & q_{33} & q_{34} & q_{35} & q_{36} & -a_{13} & -a_{23} & -a_{33} & -\mu_{13} & -\mu_{23} & -\mu_{33}
\end{bmatrix} \tag{32}$$

Other terms in Eq. (28) include β, which is a 12×1 matrix containing the second-order thermal stress tensor Λ_{ij}, the vector of pyroelectric p_i, and the vector of pyromagnetic m_i expressed as

$$\beta = \lfloor \Lambda_{11} \;\; \Lambda_{22} \;\; \Lambda_{33} \;\; \Lambda_{23} \;\; \Lambda_{13} \;\; \Lambda_{12} \;\; p_1 \;\; p_2 \;\; p_3 \;\; m_1 \;\; m_2 \;\; m_3 \rfloor^T \tag{33}$$

The coefficient in the last term c_v is the specific heat per unit volume at constant strain. Note in the input, we actually input CTE α_{kl} to be consistent with what has been normally used in thermoelastic analyses. The code automatically computes Λ_{ij} according to the following formula:

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

Note in the right hand side, the off-diagonal CTEs are multiplied by 2 so that we have $2\alpha_{12}$, $2\alpha_{13}$, $2\alpha_{23}$ according to the engineering notation.

As it is a constant confusion among users regarding the units used in the multiphysics modeling, we will provide a detailed description of those units. According to the International Standard unit system, we use **Pa** (*i.e.*, $\mathbf{N/m^2}$) for the elastic constants C_{ijkl} and the stress field σ_{ij} (note the strain field ε_{ij} is unitless), $\mathbf{C/m^2}$ for piezoelectric constants e_{ijk} and electric displacement D_i, $\mathbf{N/(A{\cdot}m)}$ for piezomagnetic constants q_{ijk} and magnetic induction B_i, $\mathbf{C/(V{\cdot}m)}$ for dielectric constants k_{ij}, $\mathbf{N/A^2}$ (or $\mathbf{N{\cdot}s^2/C^2}$) for magnetic permeability μ_{ij}, $\mathbf{C/(A{\cdot}m)}$ for electromagnetic coefficients a_{ij}, $\mathbf{V/m}$ for electric field E_i, $\mathbf{A/m}$ for magnetic field H_i, $\mathbf{K}$ for the temperature field θ (note $^\circ\mathbf{C}$ has the same unit dimension as $\mathbf{K}$), $\mathbf{1/K}$ for CTE α_{ij} (correspondingly $\mathbf{Pa/K}$ for thermal stress coefficients Λ_{ij}), $\mathbf{C/m^2{\cdot}K}$ for pyroelectric constants p_i, $\mathbf{N/(A{\cdot}m{\cdot}K)}$ for pyromagnetic m_i, and $\mathbf{J/(m^3{\cdot}K)}$ for the specific heat c_v. With all these units, the energy density U will be in the unit of $\mathbf{N/m^2}$, which is the same as $\mathbf{J/m^3}$. Note $\mathbf{N{=}C{\cdot}V/m}$ and $\mathbf{J{=}N{\cdot}m}$.

Although the units aforementioned are consistent with each other, direct use of these units will introduce an extremely ill-conditioned material matrix L as for regular materials, we have C_{ijkl} in the order of 10^{11}, while k_{ij} in the order of 10^{-9}. Proper scaling is needed even if double precision is used in computing. To this end, we define $E_i^* = \frac{E_i}{10^9}, H_i^* = \frac{H_i}{10^9}$, then the energy density in Eq. (28) can be rewritten as:

$$\frac{U}{10^9} = \frac{1}{2}\begin{Bmatrix}\varepsilon\\-E^*\\-H^*\end{Bmatrix}^T\begin{bmatrix}C^* & e & q\\ e^T & -k^* & -a^*\\ q^T & -a^{*T} & -\mu^*\end{bmatrix}\begin{Bmatrix}\varepsilon\\-E^*\\-H^*\end{Bmatrix} + \begin{Bmatrix}\varepsilon\\-E^*\\-H^*\end{Bmatrix}^T\begin{Bmatrix}-C^*\alpha\\ p\\ m\end{Bmatrix}\theta - \int_{T_1}^{T}\int_{T_1}^{\zeta}\frac{c_v^*(0,\rho)}{\rho}d\rho d\zeta \tag{35}$$

with

$$C^* = \frac{C}{10^9}, \qquad c_v^* = \frac{c_v}{10^9}, \qquad k^* = k\times 10^9, \qquad a^* = a\times 10^9, \qquad \mu^* = \mu\times 10^9 \tag{36}$$

The generalized Hooke's law given in Eq. (26) can be rewritten in the following matrix form:

$$\begin{aligned}\sigma^* &= C^*\varepsilon - eE^* - qH^* + \Lambda^*\theta\\ D &= e^T\varepsilon + k^*E^* + a^*H^* + p\theta\\ B &= q^T\varepsilon + a^{*T}E^* + \mu^*H^* + m\theta\end{aligned} \tag{37}$$

with $\sigma^* = \frac{\sigma}{10^9}$. For SwiftComp$^{\text{TM}}$ to perform multiphysics homogenization, we input C^*, c_v^*, e, q, k^*, a^*, μ^*, α, p, m as material properties, and for SwiftComp$^{\text{TM}}$ to perform multiphysics dehomogenization, we input ε, E^*, H^* as the global fields. In other words, if the quantities are given in IS units, we need to divide C, c_v, E, H by 10^9, and multiply k, a, μ by 10^9, and all the other quantities remain the same. The output effective properties are also scaled the same way as the input material properties. As far as the local fields out of dehomogenization are concerned, the mechanical displacement, strains, electric displacements, and magnetic induction are the same as SI units, however one needs to multiply the electromagnetic potential, the stresses, electric and magnetic fields with 10^9 to convert these quantities in SI units. Note, it is just one suggestion for users to scale SwiftComp$^{\text{TM}}$ inputs to avoid numerical difficulties. This scaling is done externally by the end user of the code. One can certainly devise a different scaling following the same idea given here.

7 SwiftComp™ Execution

SwiftComp™ along with a simple graphic user interface (GUI) based on Gmsh[17] is available on cdmHUB (https://cdmhub.org/resources/scstandard). To model Textile composites, SwiftComp™ is integrated with TexGen[18] which is also available on cdmHUB (https://cdmhub.org/resources/texgen4sc). One only needs a browser connected to the Internet to execute these two codes. One can also request a copy of SwiftComp™ installed on their own computer.

SwiftComp™ is distributed in the form of SwiftCompx.y.zReasePCMM-DD-YEAR.zip for Windows operating systems and SwiftCompx.y.zReaseLinuxMM-DD-YEAR.zip for Linux operating system with "x.y.z" denotes the version number. Unzipping the file is all you need to do for installing SwiftComp™. If you want to execute SwiftComp in a folder different from where you stored the executable, you need to set the path to point to the folder containing the SwiftComp executable. More details can refer to the Readme file.

Currently, we only provide executables for computers with Windows or Linux OS. The Gmsh-based GUI, TexGen-bsed GUI, and interfaces with other commercial FEA software packages such as Ansys, Abaqus, Nastran can also be freely downloaded from cdmHUB.org. Ansys-SwiftComp GUI can be found at https://cdmhub.org/resources/1136. Abaqus-SwiftComp GUI can be found at https://cdmhub.org/resources/1134. Nastran-SwiftComp GUI can be found at https://cdmhub.org/resources/1752

Without a graphic user interface, SwiftComp™ should be executed under command line. For example in Windows systems, using the system command *cmd* to bring up the command line window. Then use the system command *cd* to enter the right folder where the input files are in. Then type *SwiftComp arg1 arg2 arg3 arg4*, where *SwiftComp* is the command name of SwiftComp™, *arg1* is the complete input file name (including the extension), *arg2* denotes the macroscopic model to be constructed (*1D* for a beam model, *2D* for a plate/shell model, and *3D* for a 3D model). *arg3* denotes whether homogenization, dehomogenization, or failure analysis will be carried out with *H* for homogenization, *L* for dehomogenization, *LG* for dehomogenization with local results written in Gmsh format, *F* for initial failure strength analysis, *FE* for initial failure envelope, *FI* for initial failure indexes and strength ratios, *HA* for homogenization of aperiodic structures, *LA* for dehomogenization of aperiodic structures, and *LAG* for dehomogenization of aperiodic structures with local results written in Gmsh format. *arg4* denotes whether to use reduced integration for certain elements: *R* for reduced integration, no arguments for full integration. For example, one wants to construct a 3D model using the input file *test.sc*, then the command *SwiftComp test.sc 3D H* should be used for a homogenization run. If local fields are also desired, the command *SwiftComp test.sc 3D L* should be used for a dehomogenization run. Note for a specific input file, dehomogenization and initial failure related analyses (*arg3=F, FE, or FI*) can only be carried out after at least one homogenization run and a corresponding file with extension **.glb* storing extra inputs needed for dehomogenization and initial failure related analyses should also exist in the same folder. For example for the input file *test.sc*, we should have also prepared a file named *test.sc.glb* with details described later for a successful dehomogenization or initial failure related analyses run.

SwiftComp™ inputs are in free format which means that the entries in input files can be

separated by comma, space, and/or tab, scientific notation such as $1.0E-5$ can be used. Since SwiftComp$^{\text{TM}}$ calculation is carried out using double precision, it can also handle inputs up to 15 significant digits.

8 SwiftComp$^{\text{TM}}$ Inputs for Homogenization Run

Although general-purpose preprocessors can been developed to prepare SwiftComp$^{\text{TM}}$ input files, it is still beneficial for advanced users, particularly those who want to embed SwiftComp$^{\text{TM}}$ in their own software environments, to understand the meaning of the input data.

8.1 Extra Inputs for Dimensionally Reducible Structures

To construct a beam/plate/shell model, the beginning of the input file has two extra lines for a plate/shell model, and three extra lines for a beam model. The first line of the input file is an integer to denote a specific model. If it is 0, it will construct a classical model (Euler-Bernoulli beam model or Kirchhoff-Love plate/shell model). If it is 1, it will construct a shear refined model (Timoshenko beam model or Reissner-Mindlin plate/shell model). For a beam model, it can also be 2, indicating a Vlasov beam model, or 3 indicating a beam model with the trapeze effect.

The next line has three real numbers (k_{11}, k_{12}, k_{13} for beams) or two real numbers (k_{12}, k_{21} for shells) as the initial twist/curvatures of the structure. If the structure is initially straight, zeroes should be provided instead.

To construct a beam model, two real numbers will be additionally provided on the third line as the parameters to specify an SG which is an oblique cross-section, see Figure 13 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 13. For normal cross-sections, we provide 1.0 0.0 on this line instead.

8.2 Inputs for All Structural Models

The following line (note: to construct a 3D structural model, the previous three lines do not exist and the input file starts from this line.) has four integers providing the problem control parameters as:

analysis elem_flag trans_flag temp_flag

The parameter *analysis* is an integer denoting the type of analysis: 0-elastic; 1-thermoelastic; 2-conduction; 3-piezoeletric/piezomagnetic; 4-thermopiezoeletric/thermopiezomagnetic; 5-piezoeletromagnetic; 6-thermopiezoeletromagnetic; 7-viscoelastic; 8-thermoviscoelastic; 9-homogenization to

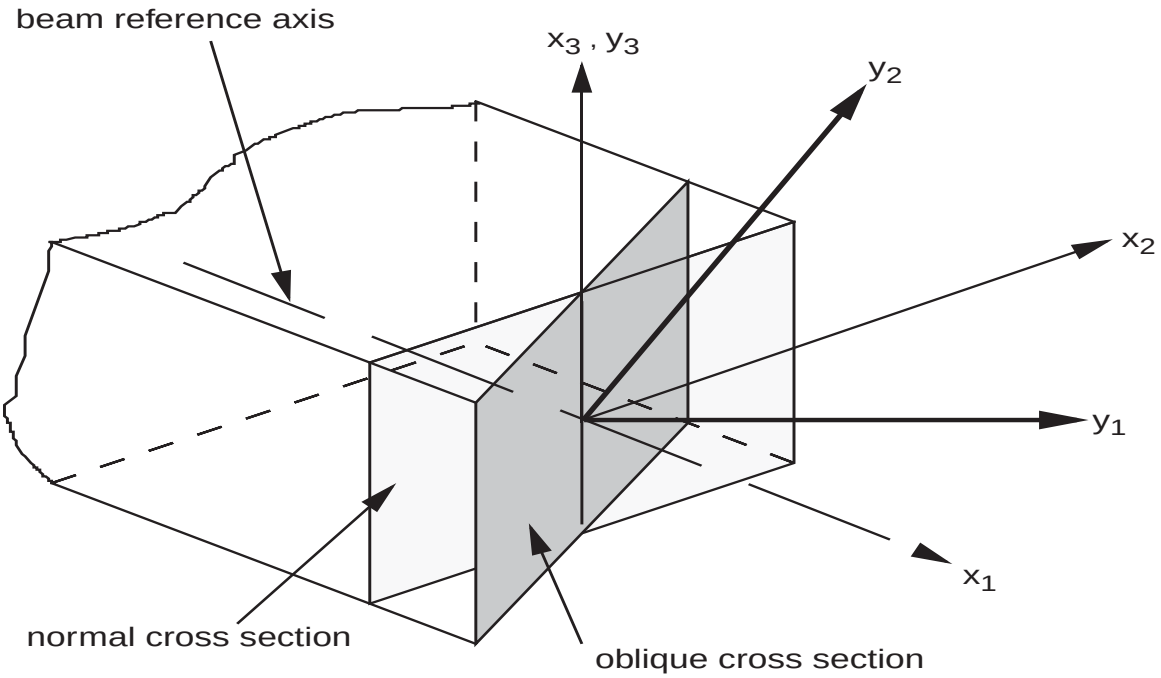

Figure 13: Sketch of an oblique reference cross-section.

8-node 3D element; 10- homogenization to 20-node 3D element. It is pointed out here that piezoelectric effects are mathematically equivalent to piezomagnetic effects. In other words, the same equation or code used for modeling piezoelectric materials can be used to model piezomagnetic materials if we replace electric displacement D_i with magnetic induction B_i, electric field E_i with magnetic field H_i, piezoelectric properties e_{kij} with piezomagnetic properties q_{kij}, pyroelectric properties p_i with pyromagnetic properties m_i. Later, for *analysis=3* or 4, in the inputs we used piezoelectric materials as example. It is directly applicable to piezomagnetic materials.

The parameter *elem_flag* is an integer denoting the type of elements. If *elem_flag* is equal to 0, the regular elements as shown in Figures 8, 9, 10 will be used for 1D, 2D or 3D SGs. If it is equal to 1, elements with one dimension degenerated will be used to model the SG. For example, 2D shell elements based on relative degrees of freedom will be used to mesh a 3D SG or 1D elements will be used to mesh a 2D SG. If it is equal to 2, 1D elements will be used to mesh a 3D SG. For example, 1D beam elements can be used to model a 3D SG composed of slender truss-like members. Currently only regular elements are implemented.

The parameter *trans_flag* is an integer denoting whether transformation of the element orientation is needed. If *trans_flag* is equal to 0, element orientation is the same as the problem coordinate system and transformation is not needed. If it is equal to 1, elemental coordinate systems are defined for each element and elemental orientations will be provided in a later block for the transformation.

The parameter *temp_flag* is an integer denoting whether the temperature is uniform within the SG. For thermally coupled analysis (*analysis=1, 4, 6*), if *temp_flag* is equal to 0, temperature distribution within SG is uniform; if it is equal to 1, temperature distribution is not uniform and nodal temperature should be provided to describe the temperature field. Note this input is used only if it is a thermally coupled analysis.

If *analysis=7* or 8, the next line will list three real numbers arranged as:

t_0, t_e, dt

where t_0 is the starting time, t_e is the ending time, and dt is the increment of time. Note, in the current version we follow the conventional practice of thermoviscoelastic analysis. The real time ranges from 10^{t_0} to 10^{t_e} with time increment of 10^{dt}. The code will compute time-dependent effective properties at $10^{t_0}, 10^{t0+dt}, \ldots, 10^{t_e}$.

If the SG is aperiodic or partially periodic, the next line will list three integers arranged as:

py1 py2 py3

where $py1$, $py2$ $py3$ could be 1, or 0 indicating whether it is aperiodic or periodic along y_1, y_2, y_3 directions, respectively. For example, for a 3D SG which is aperiodic along y_2 direction, we will have

0 1 0

It noted that for the 2D plate/shell model, only y_1 or y_2 can be periodic or aperiodic and for the 1D beam model, only y_1 can be periodic or aperiodic.

The next line lists six integers arranged as:

nSG nnode nelem nmate nslave nlayer nsurf_nodes

where *nSG* is the dimensionality of the SG, *nnode* is the total number of nodes, *nelem* is the total number of elements, *nmate* is the total number of material types, *nslave* is the number of slave nodes on periodic boundaries for periodic microstructures, and *nlayer* is the total number of layers defined by different types of materials and layup angle. *nsurf_nodes* is the total number of nodes for the outside surfaces of the SG. For the current version, *nsurf_nodes* is only implemented for *analysis=9* or 10. If *nslave=0*, SwiftComp$^{\text{TM}}$ will search for corresponding node on periodic boundaries. For this reason, the SG must be regular rectangles (for 2D SG) or cuboids (for 3D SG) with the FE mesh having corresponding nodes on periodic edges. For other periodic SG shapes, the paired nodes on corresponding boundaries must be provided through setting *nslave* not equal to zero.

The next *nnode* lines are the coordinates for each node arranged as:

node_no y_1 y_2 y_3

where *node_no* is an integer representing the unique number assigned to each node and y_1, y_2, y_3 are three real numbers describing the location (y_1, y_2, y_3) of the node (only y_3 exists for 1D SGs, and y_2 and y_3 exist for 2D SGs). Arrangement of *node_no* is not necessary to be consecutive, but all nodes from *1* to *nnode* should be present.

The next *nelem* lines list the layer number and nodes for each element. They are arranged as:

elem_no mate_id node_1 node_2 ...

where *elem_no* is the number of element, *mate_id* is an integer to indicate the material number of

the element, and *node_i* ($i = 1, 2, \ldots,$) are nodes belonging to this element. If *nlayer* is not equal to zero, then *mate_id* should be replaced with *layer_id* which will be defined later. Arrangement of *elem_no* is not necessary to be consecutive, but all elements starting from *1* to *nelem* should be present. If the SG is meshed using regular elements (*e.g.*, elements having the same dimension as the SG,

- 1D elements could have up to 5 nodes. If a node is not present in the element, the value is 0; see Figure 8.
- 2D elements could have up to 9 nodes. If a node is not present in the element, the value is 0. If the fourth node is zero, it is a triangular element; see Figure 9.
- 3D elements could have up to 20 nodes. If a node is not present in the element, the value is 0. If the fifth node is zero, it is a tetrahedral element; If the fifth node is not zero, but the seventh node is zero, it is a wedge element; see Figure 10.

If *trans_flag* is equal to 1, the next *nelem* lines list the orientation for each element. They are arranged as

elem_no a_1 a_2 a_3 b_1 b_2 b_3 c_1 c_2 c_3

where *elem_no* is the number of element, a_1, a_2, a_3 are coordinates of point a, b_1, b_2, b_3 are coordinates of point b, c_1, c_2, c_3 are coordinates of point c. The local coordinate system for the element is defined by the three points a, b, c as described previously. Arrangement of *elem_no* is not necessary to be consecutive, but all elements starting from *1* to *nelem* should be present.

If *temp_flag* is equal to 1, the temperature distribution within the SG is not uniform, the next *nnode* lines list the corresponding nodal temperature. They are arranged as:

node_no T

where *node_no* is the nodal number and T is the corresponding temperature.

If *analysis* is equal to 9, the next line lists the nodes of the macroscopic 3D 8-node element. They are arranged as:

node_1 node_2 node_3 ... node_8

where $node_i$ corresponds to the 8 nodes of the macroscopic 3D 8-node element numbered in the same order as those in Figure 10.

If *analysis* is equal to 10, the next line lists the nodes of the macroscopic 3D 20-node element. They are arranged as:

node_1 node_2 node_3 ... node_20

where $node_i$ corresponds to the 20 nodes of the macroscopic 3D 20-node element numbered in the same order as those in Figure 10.

If *analysis* is equal to 9 or 10, the next one or more lines list the nodes on the surfaces. The total number of surface nodes is *nsurf_nodes*. No specific format is needed.

If *nslave* is not equal to 0, the next *nslave* lines list the slave nodes and corresponding master nodes periodic to the slave nodes. They are arranged as:

slave_node master_node

where *slave_node* is an integer indicating the node slaved to the master node denoted by *master_node*.

If *nlayer* is not equal to 0, the next *nlayer* lines list the definition for each layer. They are arranged as:

layer_id mate_id angle

where *layer_id* is the layer number, *mate_id* is the material type, and *angle* is a real number for the layup angle in degrees.

The next *nmate* blocks define the material properties. They are arranged as:

mat_id isotropy ntemp

where *mat_id* is the material type, and *isotropy* is an integer to indicate whether the material is isotropic (0), orthotropic (1), or general anisotropic (2). The integer *ntemp* is number of material property sets according to different temperature followed by *ntemp* blocks of real numbers:

T_i ρ

const1 const2 ....

where $T_i, i = 1, \ldots, ntemp$ is the temperature, ρ is the density, and the rest are material constants the details of which will be given in the next section. For the convenience to compute secant values contained in thermal properties, T_1 is used as the reference temperature and T_i must be arranged in an increasing fashion.

If *analysis=7* or *8*, the material properties could be provided by time dependent functions. The first line of the block will become:

T_i ρ *time_function nprony*

where *time_function* is a character and *nprony* is an integer. If *time_function* is C, properties do not depend on time and *nprony* can be an arbitrary number and *nprony* is not used in the code. This line will be followed by material constants input in the same way as for other analysis options. If *time_function=P* or T, *nprony* denotes the total number of property sets used to define the time dependent properties. If *time_function=P*, the time dependency is defined using the Prony series and *nprony* is number of Prony series terms plus 1 because the long term properties are also input in the first block. If *time_function=T*, the time dependency is defined by a user defined time function with linear interpolation between adjacent time values. The material block followed by *nprony* sets of material properties arranged as

t_i

const1 const2 ....

where t_i is a real number indicating the real time if *time_function=T* or the relaxation constants if *time_function=P*. If *time_function=P*, t_1 can be arbitrary because it is followed by the long term properties and the corresponding t_1 is not used in computing the time-dependent material properties.

8.3 Conductivity

For **conduction analysis** (*analysis=2*), if the material is isotropic (*isotropy=0*), there is only one constant specifying the conductivity arranged as:

k

If *isotropy=1* (orthotropic), there are three constants arranged as:

k_{11} k_{22} k_{33}

where k_{11}, k_{22}, k_{33} are conductivities along three principal directions.

If *isotropy=2* (general anisotropic), there are six constants arranged as:

$$\begin{matrix} k_{11} & k_{12} & k_{13} \\ & k_{22} & k_{23} \\ & & k_{33} \end{matrix}$$

where k_{ij} are the components of the second-order conductivity tensor.

8.4 Elastic Properties

For all other analyses, we need to first provide elastic properties. If *isotropy=0*, there are two constants arranged as:

E ν

where E is the Young's modulus, ν is the Poisson's ratio. Elasticity theory restricts that Poisson's ratio must be greater than -1.0 and less than 0.5 for linear, elastic, isotropic materials, although SwiftComp$^{\text{TM}}$ allows users to input values that are very close to those limits.

If *isotropy=1*, there are nine constants arranged as:

E_1 E_2 E_3

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

ν_{12} ν_{13} ν_{23}

including Young's moduli (E_1, E_2, and E_3), shear moduli (G_{12}, G_{13}, and G_{23}), and Poisson's ratios (ν_{12}, ν_{13}, and ν_{23}). The convention of values is such that these values will be used to form the Hooke's law for the orthotropic material in Eq. (6).

If *isotropy=2*, there are 21 constants arranged as:

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

These values are defined using the Hooke's law given in Eq. (3). For orthotropic and general anisotropic materials, SwiftComp$^{\text{TM}}$ does not check the validity of the material inputs and it is the user's responsibility to make sure that the material properties satisfy the positive definiteness requirement of the stiffness matrix and compliance matrix found in a typical textbook on mechanics

of composite materials.

8.5 CTE and Specific Heat

For thermally coupled analysis (*analysis=1, 4, 6*), if *isotropy=0*, there are two constants arranged as:

$\alpha \quad c_v$

where α is the CTE and c_v is the specific heat.

If *isotropy=1*, there are four constants arranged as:

$\alpha_{11} \quad \alpha_{22} \quad \alpha_{33} \quad c_v$

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

If *isotropy=2*, there are seven constants arranged as:

$\alpha_{11} \quad \alpha_{22} \quad \alpha_{33} \quad 2\alpha_{23} \quad 2\alpha_{13} \quad 2\alpha_{12} \quad c_v$

where α_{ij} are the components of the second-order CTE tensor.

8.6 Piezoelectric, Dielectric, and Pyroelectric Coefficients

If *analysis=3,4,5,6*, we then need to provide 18 piezoelectric coefficients arranged as:

$e_{11} \quad e_{12} \quad e_{13} \quad e_{14} \quad e_{15} \quad e_{16}$
$e_{21} \quad e_{22} \quad e_{23} \quad e_{24} \quad e_{25} \quad e_{26}$
$e_{31} \quad e_{32} \quad e_{33} \quad e_{34} \quad e_{35} \quad e_{36}$

The piezoelectric coefficients should be followed by dielectric properties. If *isotropy=0*, there is one constant for dielectric coefficient arranged as:

k

If *isotropy=1*, there are three constants arranged as:

$k_{11} \quad k_{22} \quad k_{33}$

where k_{11}, k_{22}, k_{33} are the dielectric coefficients along three principal directions.

If *isotropy=2*, there are six constants arranged as:

$k_{11} \quad k_{12} \quad k_{13}$
$\quad\quad k_{22} \quad k_{23}$
$\quad\quad\quad\quad k_{33}$

where k_{ij} denotes the second-order dielectric tensor.

If *analysis=4,6*, the above data should be followed by three pyroelectric coefficients arranged as:

$p_1 \quad p_2 \quad p_3$

8.7 Piezomagnetic Coefficients, Magnetic Permeability, Electromagnetic and Pyromagnetic Coefficients

To carry out coupled piezoelectromagnetic analysis (*analysis=5,6*), we then need to provide another 18 piezomagnetic coefficients arranged as:

q_{11} q_{12} q_{13} q_{14} q_{15} q_{16}
q_{21} q_{22} q_{23} q_{24} q_{25} q_{26}
q_{31} q_{32} q_{33} q_{34} q_{35} q_{36}

The piezomagnetic coefficients should be followed by magnetic permeability and electromagnetic coefficients. If *isotropy=0*, there are two constants arranged as:

μ
a

with μ as the magnetic permeability, and a as the electromagnetic coefficient.

If *isotropy=1*, there are six constants arranged as:

μ_{11} μ_{22} μ_{33}
a_{11} a_{22} a_{33}

where $\mu_{11}, \mu_{22}, \mu_{33}$ are the magnetic permeability along three principal directions, and a_{11}, a_{22}, a_{33} are the electromagnetic coefficients along three principal directions.

If *isotropy=2*, there are 12 constants arranged as:

μ_{11} μ_{12} μ_{13}
μ_{22} μ_{23}
μ_{33}
a_{11} a_{12} a_{13}
a_{22} a_{23}
a_{33}

where μ_{ij} denotes the second-order magnetic permeability tensor and a_{ij} denotes the second-order electromagnetic coupling tensor.

If *analysis=6*, the above data should be followed by three pyromagnetic coefficients arranged as:

m_1 m_2 m_3

To clarify the order of input material properties, if *analysis=6* and *isotropy=2*, the material inputs should be arranged:

T_i ρ
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}

α_{11} α_{22} α_{33} $2\alpha_{23}$ $2\alpha_{13}$ $2\alpha_{12}$ c_v
e_{11} e_{12} e_{13} e_{14} e_{15} e_{16}
e_{21} e_{22} e_{23} e_{24} e_{25} e_{26}
e_{31} e_{32} e_{33} e_{34} e_{35} e_{36}
k_{11} k_{12} k_{13}
k_{22} k_{23}
k_{33}
p_1 p_2 p_3
q_{11} q_{12} q_{13} q_{14} q_{15} q_{16}
q_{21} q_{22} q_{23} q_{24} q_{25} q_{26}
q_{31} q_{32} q_{33} q_{34} q_{35} q_{36}
μ_{11} μ_{12} μ_{13}
μ_{22} μ_{23}
μ_{33}
a_{11} a_{12} a_{13}
a_{22} a_{23}
a_{33}
m_1 m_2 m_3

The material constants could be expressed either in the problem coordinate system y_1, y_2, y_3 or in the material coordinate system. However, it is usually more convenient and simpler to provide these constants in the material coordinate system. If it is expressed in the material coordinate system, SwiftComp$^{\text{TM}}$ will perform the necessary transformations. The input quantities should be properly scaled as discussed previously for multiphysics modeling. It is also emphasized that if the users uses an arrangement of stresses and strains different from what SwiftComp$^{\text{TM}}$ uses, proper re-arrangement of the material properties is needed.

The following line is used to input ω, the volume of the domain spanned by the remaining coordinates in the macroscopic structural model. For 3D structural models, ω will be the volume of the homogenized material including both the volume of the material and the volume of possible voids in the SG. ω can be computed by any mesh generator. For regular SG such as cubes, it can be easily calculated by hand. Of course, for 1D SGs, the volume is the length and for 2D SGs, the volume is the area. For plate/shell models, ω will be the area spanned by y_1 and y_2 for 3D SGs, the length along y_2 for 2D SGs and 1.0 for 1D SGs. For beam models, ω will be the length along y_1 for 3D SGs and 1.0 for 2D SGs.

Till now, we have prepared all the inputs necessary for the homogenization run.

9 SwiftComp$^{\text{TM}}$ Inputs for Dehomogenization Run

For dehomogenization, the user needs to provide additional information obtained from the macroscopic analysis including the macroscopic primary field (such as temperature for heat conduction or displacement for the elastic analysis) and the generalized strain vector according to Eq. (29). This information are provided in a text file corresponding to the input file name with extension *glb*.

For example if the input file is *test.sc*, one should also prepare a file called *test.sc.glb* for dehomogenization holding the data as described below.

If *analysis= 0*, then the macroscopic displacements, rotations, and mechanical strains are to be provided to compute the local displacement/strain/stress fields. The data are arranged as:

v_1 v_2 v_3
C_{11} C_{12} C_{13}
C_{21} C_{22} C_{23}
C_{31} C_{32} C_{33}
id_1
$\bar{\epsilon}$ or $\bar{\sigma}$

where v_1, v_2, and v_3 are the macro displacements, C_{ij} are macro rotations, and $\bar{\epsilon}$ contains the macro generalized strains. C_{ij} is defined such that $\boldsymbol{B}_i = C_{ij}\boldsymbol{b}_j$ where $\boldsymbol{b}_j$ is the base vector for undeformed configuration and $\boldsymbol{B}_i$ is the base vector for the deformed configuration. For example, for a linear analysis of 3D structures,

$$C_{ij} = \begin{bmatrix} 1+u_{1,1} & u_{2,1} & u_{3,1} \\ u_{1,2} & 1+u_{2,2} & u_{3,2} \\ u_{1,3} & u_{2,3} & 1+u_{3,3} \end{bmatrix} \tag{38}$$

For linear analysis of plates/shells using the classical plate model (Kirchhoff-Love model),

$$C_{ij} = \begin{bmatrix} 1+u_{1,1} & u_{2,1} & u_{3,1} \\ u_{1,2} & 1+u_{2,2} & u_{3,2} \\ -u_{3,1} & -u_{3,2} & 1+u_{1,1}+u_{2,2} \end{bmatrix} \tag{39}$$

For linear analysis of beams using the classical beam model (Euler-Bernoulli model),

$$C_{ij} = \begin{bmatrix} 1 & u_2' & u_3' \\ -u_2' & 1 & \theta_1 \\ -u_3' & -\theta_1 & 1 \end{bmatrix} \tag{40}$$

where u_i are the global displacements and θ_1 is the twist angle.

Here id_1 indicates whether generalized stresses or generalized strains are used for dehomogenization run. If it is equal to 0, generalized stresses are used as inputs; if it is equal to 1, generalized strains are used as inputs. For the 3D Cauchy continuum model $\bar{\epsilon} = \lfloor \epsilon_{11} \ \epsilon_{22} \ \epsilon_{33} \ 2\epsilon_{23} \ 2\epsilon_{13} \ 2\epsilon_{12} \rfloor^T$, $\bar{\sigma} = \lfloor \sigma_{11} \ \sigma_{22} \ \sigma_{33} \ \sigma_{23} \ \sigma_{13} \ \sigma_{12} \rfloor^T$. For the Kirchhoff-Love plate/shell model, $\bar{\epsilon} = \lfloor \epsilon_{11} \ \epsilon_{22} \ 2\epsilon_{12} \ \kappa_{11} \ \kappa_{22} \ 2\kappa_{12} \rfloor^T$, $\bar{\sigma} = \lfloor N_{11} \ N_{22} \ N_{12} \ M_{11} \ M_{22} \ M_{12} \rfloor^T$. For the Reissner-Mindlin plate/shell model, $\bar{\epsilon} = \lfloor \epsilon_{11} \ \epsilon_{22} \ 2\epsilon_{12} \ \kappa_{11} \ \kappa_{22} \ 2\kappa_{12} \ \gamma_{13} \ \gamma_{23} \rfloor^T$, $\bar{\sigma} = \lfloor N_{11} \ N_{22} \ N_{12} \ M_{11} \ M_{22} \ M_{12} \ N_{13} \ N_{23} \rfloor^T$. For the Euler-Bernoulli beam model, $\bar{\epsilon} = \lfloor \epsilon_{11} \ \kappa_{11} \ \kappa_{12} \ \kappa_{13} \rfloor^T$, $\bar{\sigma} = \lfloor F_1 \ M_1 \ M_2 \ M_3 \rfloor^T$. For the Timoshenko beam model, $\bar{\epsilon} = \lfloor \epsilon_{11} \ \gamma_{12} \ \gamma_{13} \ \kappa_{11} \ \kappa_{12} \ \kappa_{13} \rfloor^T$, $\bar{\sigma} = \lfloor F_1 \ F_2 \ F_3 \ M_1 \ M_2 \ M_3 \rfloor^T$.

If *analysis=1*, we need to provide an additional data for the macroscopic temperature difference T_m to compute the thermoelastic effects. T_m is the difference between the current macroscopic temperature with respect to the reference temperature T_1. If *temp_flag=1*, T_m is not used. The data are arranged as:

v_1 v_2 v_3

C_{11} C_{12} C_{13}
C_{21} C_{22} C_{23}
C_{31} C_{32} C_{33}
id_1
$\bar{\epsilon}$ or $\bar{\sigma}$
T_m

If *analysis=2*, we need to provide the following four values, arranged as
T
id_1
$T_{,1}$ $T_{,2}$ $T_{,3}$ or $-q_1$ $-q_2$ $-q_3$
where $T_{,i}$ are the gradients of the macroscopic temperature, and q_i are the macroscopic heat fluxes. If $id_1 = 0$, heat fluxes are used as inputs. If $id_1 = 1$, temperature gradients are used as inputs.

If *analysis= 3* or 4, we need to provide the following data for dehomogenization, which are arranged as:
v_1 v_2 v_3 ϕ^*
C_{11} C_{12} C_{13}
C_{21} C_{22} C_{23}
C_{31} C_{32} C_{33}
id_1
$\bar{\epsilon}$ or $\bar{\sigma}$
T_m
with ϕ^* as the scaled electric potential, $\bar{\epsilon}$ as the macro generalized strains, and $\bar{\sigma}$ as the macro generalized stresses. For 3D structures $\bar{\epsilon} = \lfloor \epsilon_{11} \ \epsilon_{22} \ \epsilon_{33} \ 2\epsilon_{23} \ 2\epsilon_{13} \ 2\epsilon_{12} \ -E_1^* \ -E_2^* \ -E_3^* \rfloor^T$, $\bar{\sigma} = \lfloor \sigma_{11} \ \sigma_{22} \ \sigma_{33} \ \sigma_{23} \ \sigma_{13} \ \sigma_{12} \ D_1 \ D_2 \ D_3 \rfloor^T$. For the Kirchhoff-Love plate/shell model, $\bar{\epsilon} = \lfloor \epsilon_{11} \ \epsilon_{22} \ 2\epsilon_{12} \ \kappa_{11} \ \kappa_{22} \ 2\kappa_{12} \ -E_1^* \ -E_2^* \rfloor^T$, $\bar{\sigma} = \lfloor N_{11} \ N_{22} \ N_{12} \ M_{11} \ M_{22} \ M_{12} \ D_1 \ D_2 \rfloor^T$. For the Reissner-Mindlin plate/shell model, $\bar{\epsilon} = \lfloor \epsilon_{11} \ \epsilon_{22} \ 2\epsilon_{12} \ \kappa_{11} \ \kappa_{22} \ 2\kappa_{12} \ \gamma_{13} \ \gamma_{23} \ -E_1^* \ -E_2^* \rfloor^T$, $\bar{\sigma} = \lfloor N_{11} \ N_{22} \ N_{12} \ M_{11} \ M_{22} \ M_{12} \ N_{13} \ N_{23} \ D_1 \ D_2 \rfloor^T$. For the Euler-Bernoulli beam model, $\bar{\epsilon} = \lfloor \epsilon_{11} \ \kappa_{11} \ \kappa_{12} \ \kappa_{13} \ -E_1^* \rfloor^T$, $\bar{\sigma} = \lfloor F_1 \ M_1 \ M_2 \ M_3 \ D_1 \rfloor^T$. For the Timoshenko beam model, $\bar{\epsilon} = \lfloor \epsilon_{11} \ \gamma_{12} \ \gamma_{13} \ \kappa_{11} \ \kappa_{12} \ \kappa_{13} \ D_1 \rfloor^T$, $\bar{\sigma} = \lfloor F_1 \ F_2 \ F_3 \ M_1 \ M_2 \ M_3 \ D_1 \rfloor^T$. Here $\phi^*_{,i} = -E_i^*$. If *temp_flag=1*, T_m is not used. If *analysis* is 3, the macroscopic temperature difference T_m does not exist.

If *analysis= 5* or 6, we need to provide the following data which are arranged as:
v_1 v_2 v_3 ϕ^* ψ^*
C_{11} C_{12} C_{13}
C_{21} C_{22} C_{23}
C_{31} C_{32} C_{33}
id_1
$\bar{\epsilon}$ or $\bar{\sigma}$
T_m
with ψ^* as the scaled magnetic potential, $\bar{\epsilon}$ as the macro generalized strains, and $\bar{\sigma}$ as the macro generalized stresses. For 3D structures $\bar{\epsilon} = \lfloor \epsilon_{11} \ \epsilon_{22} \ \epsilon_{33} \ 2\epsilon_{23} \ 2\epsilon_{13} \ 2\epsilon_{12} \ -E_1^* \ -E_2^* \ -E_3^* \ -$

$H_1^* \quad -H_2^* \quad -H_3^*\rfloor^T$, $\bar{\sigma} = \lfloor \sigma_{11} \quad \sigma_{22} \quad \sigma_{33} \quad \sigma_{23} \quad \sigma_{13} \quad \sigma_{12}\ D_1 \quad D_2 \quad D_3 \quad B_1 \quad B_2 \quad B_3\rfloor^T$. For the Kirchhoff-Love plate/shell model, $\bar{\epsilon} = \lfloor \epsilon_{11} \quad \epsilon_{22} \quad 2\epsilon_{12} \quad \kappa_{11} \quad \kappa_{22} \quad 2\kappa_{12} \quad -E_1^* \quad -E_2^* \quad -H_1^* \quad -H_2^*\rfloor^T$, $\bar{\sigma} = \lfloor N_{11} \quad N_{22} \quad N_{12} \quad M_{11} \quad M_{22} \quad M_{12} \quad D_1 \quad D_2 \quad B_1 \quad B_2\rfloor^T$. For the Reissner-Mindlin plate/shell model, $\bar{\epsilon} = \lfloor \epsilon_{11} \quad \epsilon_{22} \quad 2\epsilon_{12} \quad \kappa_{11} \quad \kappa_{22} \quad 2\kappa_{12} \quad \gamma_{13} \quad \gamma_{23} \quad -E_1^* \quad -E_2^* \quad -H_1^* \quad -H_2^*\rfloor^T$, $\bar{\sigma} = \lfloor N_{11} \quad N_{22} \quad N_{12} \quad M_{11} \quad M_{22} \quad M_{12} \quad N_{13} \quad N_{23} \quad D_1 \quad D_2 \quad B_1 \quad B_2\rfloor^T$. For the Euler-Bernoulli beam model, $\bar{\epsilon} = \lfloor \epsilon_{11} \quad \kappa_{11} \quad \kappa_{12} \quad \kappa_{13} \quad -E_1^* \quad -H_1^*\rfloor^T$, $\bar{\sigma} = \lfloor F_1 \quad M_1 \quad M_2 \quad M_3 \quad D_1 \quad B_1\rfloor^T$. For the Timoshenko beam model, $\bar{\epsilon} = \lfloor \epsilon_{11} \quad \gamma_{12} \quad \gamma_{13} \quad \kappa_{11} \quad \kappa_{12} \quad \kappa_{13} \quad -E_1^* \quad -H_1^*\rfloor^T$, $\bar{\sigma} = \lfloor F_1 \quad F_2 \quad F_3 \quad M_1 \quad M_2 \quad M_3 \quad D_1 \quad B_1\rfloor^T$. Here $\psi_{,i}^* = -H_i^*$. If *temp_flag=1*, T_m is not used. If *analysis* is 5, the macroscopic temperature difference T_m does not exist.

If *analysis= 9* or 10, we need to the following data which are arranged as:
$\bar{u}_{11}, \bar{u}_{12}, \bar{u}_{13}$
$\bar{u}_{21}, \bar{u}_{22}, \bar{u}_{23}$
...
$\bar{u}_{n1}, \bar{u}_{n2}, \bar{u}_{n3}$
with $\bar{u}_{i1}, \bar{u}_{i2}, \bar{u}_{i3}$ as the displacements of node i along y_1, y_2, y_3 directions respectively. It is noted that such displacements are not those measured in the global coordinate system of the macroscopic analysis, but those measured in the elemental coordinate system of the macroscopic analysis.

10 SwiftComp™ Inputs for Failure Analysis

A corresponding homogenization analysis must be run before carrying out any failure analysis.

For failure analysis (including initial failure strength, initial failure index/strength ratio, and initial failure envelope), the **.glb* file will be used to store the data needed for failure analysis instead. First, additional material properties are needed for each material at each given temperature including a failure criterion and corresponding strength constants. Three lines will be inserted and the inputs needed for failure analyses should be arranged as:
$failure\_criterion \quad num\_of\_constants$
l_c
$const_1 \quad const_2 \quad const_3 \quad \ldots$
$failure\_criterion$ is an integer identifier for the failure criterion. $num\_of\_constants$ indicates the number of strength constants needed for the corresponding failure criterion. l_c is a real number indicating the characteristic length used in the nonlocal approach for initial failure analysis. If l_c is equal to zero, the local approach based on element averaged values will be used. $const_1, const_2, const_3 \ldots$ are the corresponding strength constants. It is noted that this block of data should be corresponding to the material block in the main input file. In other words, for each material with *mat_id*, we need to provide such information for each temperature.

$failure\_criterion$ can be equal to 1, 2, 3, 4, 5, and another number greater than 10. For isotropic material, 1 is the max principal stress criterion, 2 is the max principal strain criterion, 3 is the max shear stress criterion (also commonly called the Tresca criterion), 4 is the max shear strain criterion, and 5 is the Mises criterion. For anisotropic materials, 1 is the max stress criterion for anisotropic materials, 2 is the max strain criterion for anisotropic materials, 3 is the Tsai-Hill criterion, 4 is the Tsai-Wu criterion and 5 is the Hashin criterion. 11 or a larger integer indicates

a user-defined failure criterion. If $failure\_criterion$ is equal to 1, 2, 3, 4, 5, $num\_of\_constants$ is not used. If it is a user-defined failure criterion, then $num\_of\_constants$ will be used to input the right number of strength constants. 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_ϵ, X'_ϵ.
- If $failure\_criterion$ is 3, the max shear stress criterion (aka the 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 for tensile strengths (X, Y, Z) in three directions, three for compressive strengths (X', Y', Z') in three directions, and three for 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 for tensile strengths ($X_\epsilon, Y_\epsilon, Z_\epsilon$) in three directions, three for compressive strengths ($X'_\epsilon, Y'_\epsilon, Z'_\epsilon$) in three directions, and three for shear strengths ($R_\epsilon, T_\epsilon, S_\epsilon$) in three principal planes, arranged as $X_\epsilon, Y_\epsilon, Z_\epsilon, X'_\epsilon, Y'_\epsilon, Z'_\epsilon, R_\epsilon, T_\epsilon, S_\epsilon$.
- If $failure\_criterion$ is 3, the Tsai-Hill criterion is used and six strength constants are needed: three for normal strengths (X, Y, Z) in three directions and three for shear strengths (R, S, T) 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 for tensile strengths (X, Y, Z), three for compressive strengths (X', Y', Z') in three directions, and three for 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 for tensile strengths (X, Y), two for compressive strengths (X', Y') in two directions, and two for shear strengths (R, S) in two principal planes, arranged as X, Y, X', Y', R, S.

It is noted that for failure analysis, general anisotropic materials are also approximated using orthotropic materials due to limited number of strength constants. In SwiftComp$^{\text{TM}}$, 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.

After the material block for strength parameters, we need to provide the following line in the **.glb* file containing one integer if analysis is not equal to 9 or 10.

id_1

Here id_1 indicates whether the strength is expressed in terms of generalized stresses or generalized strains. If it is equal to 0, the strength is expressed in terms of generalized stresses; if it is equal to 1, it is expressed in terms of generalized strains.

For *failure envelope analysis*, we need to provide the following line in the **.glb* file containing two integers if analysis is not equal to 9 or 10.

$istr_1$ $istr_2$

Here $istr_1$ and $istr_2$ indicate the two load directions that one would like to predict a failure envelope for. The values could be 1, 2, 3, 4,..., corresponding to the arrangement of the generalized stresses (if $id_1 = 0$) or generalized strains (if $id_1 = 1$). For example for the 3D model, the stress/strain are arranged in the order of 11, 22, 33, 23, 13, 12. If we want to draw a $\bar{\sigma}_{22}$—$\bar{\sigma}_{13}$ failure envelope, we will have $istr_1 = 2, istr_2 = 5$.

For *failure index analysis*, this above line is not necessary. Instead we need to provide the following line in the **.glb* file containing n real numbers with n equal to the total number of generalized stresses or generalized strains.

str_1 str_2 str_3 ... str_n

Here $str_1, str_2, \ldots, str_n$ indicate the given loads used to compute the strength ratio and the failure index. These values can be given in terms of generalized stresses or generalized strains depending on whether the required strength outputs are in generalized stresses ($id_1 = 0$) or generalized strains ($id_1 = 1$). If analysis is equal to 9 or 10, corresponding nodal displacements should be provided as the inputs instead as those described at the end of the previous section.

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 a file named *input_file_name.ech*. Error messages are also written at the end of *input_file_name.ech* and on the output screen.

11 SwiftComp$^{\text{TM}}$ Outputs

Effective properties computed by SwiftComp$^{\text{TM}}$ are stored in *input_file_name.k*, including effective stiffness matrix corresponding to Eq. (32), effective flexibility matrix (inverse of stiffness matrix), effective thermal coefficients including CTEs, specific heat, pyroelectric coefficients, and pyromagnetic coefficients.

If the material can be approximated as orthotropic material, engineering constants corresponding to the elastic stiffness matrix is also provided among the outputs.

Regarding the effective specific heat, there are two contributions $D_{\theta\theta}$ and F_{eff}. If *temp_flag=0*, the effective specific heat can be calculated as

$$\bar{c}_v = D_{\theta\theta} - TF_{eff}$$

with T as the current temperature $T = T_1 + T_m$. If *temp_flag=1*, the effective specific heat can be calculated as

$$\bar{c}_v = \frac{D_{\theta\theta} - T_1 F_{eff}}{\bar{\theta}^2}$$

with $\bar{\theta}$ as the average temperature of the SG.

The effective density of the SG is also listed as one output.

If *analysis=9*, the output will be a 24×24 effective element stiffness matrix for a 8-node 3D element. If *analysis=10*, the output will be a 60×60 effective element stiffness matrix for a 20-node 3D element.

In the outputs of dehomogenization, the primary local field such as the displacement field for elastic analysis or the temperature field for conduction analysis is reported at each node. However, other fields calculated based on gradients from the primary local field such as stresses and strains are usually more accurate if reported at Gaussian integration points. However, because nodal values are more convenient for postprocessing of the results, only nodal values are reported. If you need Gaussian values, please contact the author.

The local 3D displacement results obtained through dehomogenization are stored in *input_file_name.u*. The values are listed for each node identified by its location as:

node_no u_1 u_2 u_3

where u_i are the local 3D displacements at this node. Note if *analysis=2*, the outputs in this file are the local temperature for each node instead.

If *analysis=3, 4*, the outputs will be

node_no u_1 u_2 u_3 ϕ^*

with ϕ^* as the scaled electric potential.

If *analysis=5, 6*, the outputs will be

node_no u_1 u_2 u_3 ϕ^* ψ^*

with ψ^* as the scaled magnetic potential.

The local 3D strain/stress results at nodes obtained through dehomogenization are stored in *input_file_name.sn*. These values are identified by its location as:

y_1 y_2 y_3 ϵ_{11} ϵ_{22} ϵ_{33} $2\epsilon_{23}$ $2\epsilon_{13}$ $2\epsilon_{12}$ σ_{11} σ_{22} σ_{33} σ_{23} σ_{13} σ_{12}

where ϵ_{ij} and σ_{ij} are the components of the local 3D strain tensor and 3D stress tensor, respectively,

at the node. Note if *analysis=2*, the outputs in this file are instead the local temperature gradient and heat flux instead arranged as

y_1 y_2 y_3 $T_{,1}$ $T_{,2}$ $T_{,3}$ q_1 q_2 q_3.

If *analysis=3, 4*, the outputs will be

y_1 y_2 y_3 ϵ_{11} ϵ_{22} ϵ_{33} $2\epsilon_{23}$ $2\epsilon_{13}$ $2\epsilon_{12}$ $-E_1^*$ $-E_2^*$ $-E_3^*$ σ_{11} σ_{22} σ_{33} σ_{23} σ_{13} σ_{12} D_1 D_2 D_3

If *analysis=5, 6*, the outputs will be

y_1 y_2 y_3 ϵ_{11} ϵ_{22} ϵ_{33} $2\epsilon_{23}$ $2\epsilon_{13}$ $2\epsilon_{12}$ $-E_1^*$ $-E_2^*$ $-E_3^*$ $-H_1^*$ $-H_2^*$ $-H_3^*$ σ_{11} σ_{22} σ_{33} σ_{23} σ_{13} σ_{12} D_1 D_2 D_3 B_1 B_2 B_3.

If the local results are desired to be output in Gmsh format (*arg3=LG or LAG*), the recovered 3D strain/stress results at nodal points are stored in *input_file_name.sn*. This file contains a block for each generalized strain or stress component as follows:

elem_no nodes nodal_value

where *elem_no* is the corresponding element number, *nodes* is the total number of nodes in this element, and *nodal_values* is an array holding *nodes* nodal values of the strain or strain component. Each block contains *nelem* lines and blocks are separated by a blank line. For example, if *analysis=5,6*, there will be 24 blocks of data arranged according to the order first for the generalized strains, then for the generalized stresses according to SwiftComp$^{\text{TM}}$ convention specified in Eqs. (29) and (30).

The above local 3D strain/stress results are expressed in the problem coordinate system. Sometimes it is more convenient to have strain/stress values expressed in the material coordinate system. These values at nodal points are stored in the file *input_file_name.snm*.

The failure analysis results are stored in the file *input_file_name.fi*, the content of which depends on the type of analysis. For *failure index analysis (FI)*, the failure index and strength ratio for each element are stored in *input_file_name.fi* with the first number is an integer indicating the element number and the trailing two numbers are the initial failure index and the initial strength ratio for each element under given loads. For Hashin failure criterion, the failure modes are also output for the corresponding element. For *initial strength analysis (F)*, this file stores the initial failure strengths in both tensile and compressive directions. For *failure envelope analysis (FE)*, this file stores the failure envelope points with the first number being the number of the failure point, the two trailing read numbers being the values for corresponding given two loading directions needed for plotting the failure envelope.

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

12 SwiftComp$^{\text{TM}}$ Maintenance and Tech Support

SwiftComp$^{\text{TM}}$ is available in the cloud on cdmHUB and users can execute the code within a web browser. A group named *Prof. Yu's Research Group in the Cloud* is specifically set up on cdmHUB for information exchange related with SwiftComp$^{\text{TM}}$. *Users are highly encouraged to join*

the group through http://cdmhub.org/groups/yugroup to receive most recent news of SwiftComp^{TM}, ask questions, and share with others. For the sake of efficiency and reusability, **please post your questions in the group.** A page of SwiftComp^{TM} FAQ will be constantly updated in the group. Before you ask questions, please do the following:

1. Read this 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 your input file understood by SwiftComp^{TM};
4. Check the SwiftComp^{TM} FAQ page in the group;
5. Post your question in the discussion section of the group.

13 Epilogue

Although still in its early development age, SwiftComp^{TM} has demonstrated great potential for multiscale constitutive modeling of materials and structures. Its accuracy has been extensively verified by its developers and users. The performance and robustness of the code will be continuously improved based on feedback from its users throughout the world. Although SwiftComp^{TM} 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 SwiftComp^{TM} related publications), further questions are inevitable because SwiftComp^{TM} represents a unique unified approach for modeling composite structures and materials which is drastically different from most conventional micromechanics and structural mechanics approaches. Nevertheless, it should be clear that SwiftComp^{TM} is emerging as a general-purpose computational tool for engineers to perform multiscale constitutive modeling of composites.

References

[1] W. Yu. Unifying structural mechanics with micromechanics using the concept of representative structural element. In *Proceedings of the American Society for Composites 27th Technical Conference*, Arlington, Texas, Oct. 1 – 3 2012.

[2] W. Yu. Structure genome: Fill the gap between materials genome and structural analysis. In *Proceedings of the 56th Structures, Structural Dynamics and Materials Conference*, Kissimmee, Florida, Jan. 5-9 2015. AIAA.

[3] W. Yu. A unified theory for constitutive modeling of composites. *Journal of Mechanics of Materials and Structures*, 11(4):379–411, 2016.

[4] W. Yu. Simplified formulation of mechanics of structure genome. *AIAA Journal*, 57:4201–4209, 2019.

[5] S. W. Sloan. A fortran program for profile and wavefront reduction. *International Journal for Numerical Methods in Engineering*, 28:2651–2679, 1989.

[6] W. Yu. *Variational Asymptotic Modeling of Composite Dimensionally Reducible Structures*. PhD thesis, Aerospace Engineering, Georgia Institute of Technology, May 2002.

[7] C.T. Sun and R.S. Vaidya. Prediction of composite properties from a representative volume element. *Composites Science and Technology*, 56:171–179, 1996.

[8] W. Yu. Representative structural element: a new paradigm for multiscale structural modeling. In *Proceedings of the 54th Structures, Structural Dynamics and Materials Conference*, Boston, Massachusetts, Apr. 8-11 2013. AIAA.

[9] X. Liu and W. Yu. A novel approach to analyze beam-like composite structures using mechanics of structure genome. *Advances in Engineering Software*, 100:238–251, 2016.

[10] B. Peng, J. Goodsell, R. B. Pipes, and W. Yu. Generalized free-edge stress analysis using mechanics of structure genome. *Journal of Applied Mechanics*, 83(10), 2016. Article 101013.

[11] Xin Liu, Khizar Rouf, Bo Peng, , and Wenbin Yu. Two-step homogenization of textile composites using mechanics of structure genome. *Composite Structures*, 171:252–262, 2017.

[12] B. Peng and W. Yu. A micromechanics theory for homogenization and dehomogenization of aperiodic heterogeneous materials. *Composite Structures*, 199:53–62, 2018.

[13] X. Liu, T. Tang, W. Yu, and R. B. Pipes. Multiscale modeling of viscoelastic behaviors of textile composites. *International Journal of Engineering Science*, 130:175–186, 2018.

[14] X. Liu, F. Gasco, W. Yu, J. Goodsell, and K. Rouf. Multiscale analysis of woven composite structures in msc. nastran. *Advances in Engineering Software*, 135, 2019. Article 102677.

[15] X. Liu, W. Yu, and J. Gasco, F.and Goodsell. A unified approach for thermoelastic constitutive modeling of composite structures. *Composites Part B*, 172:649–659, 2019.

[16] O. Rique, X. Liu, W. Yu, and R. B. Pipes. Constitutive modeling for time- and temperature-dependent behavior of composites. *Composites Part B*, 184, 2020. Article 107726.

[17] C. Geuzaine and J.-F. Remacle. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–331, 2009.

[18] H. Lin, L. P. Brown, and A. C. Long. Modelling and simulating textile structures using texgen. *Advanced Materials Research*, 331:44–47, 2011.