Finite Element Analysis of Composite Materials Using Abaqus™

Ever J. Barbero

CRC Press
Taylor & Francis Group

Finite Element Analysis of Composite Materials Using Abaqus™

Composite Materials: Design and Analysis

Series Editor

Ever J. Barbero

PUBLISHED

Finite Element Analysis of Composite Materials with Abaqus, *Ever J. Barbero*

FRP Deck and Steel Girder Bridge Systems: Analysis and Design,
Julio F. Davalos, An Chen, Bin Zou, and Pizhong Qiao

Introduction to Composite Materials Design, Second Edition, *Ever J. Barbero*

Finite Element Analysis of Composite Materials, *Ever J. Barbero*

FORTHCOMING

Smart Composites: Mechanics and Design, *Rani El-Hajjar, Valeria La Saponara, and Anastasia Muliana*

Finite Element Analysis of Composite Materials Using Abaqus™

Ever J. Barbero

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business

MATLAB® is a trademark of The MathWorks, Inc. and is used with permission. The MathWorks does not warrant the accuracy of the text or exercises in this book. This book's use or discussion of MATLAB® software or related products does not constitute endorsement or sponsorship by The MathWorks of a particular pedagogical approach or particular use of the MATLAB® software.

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

Version Date: 20130408

International Standard Book Number-13: 978-1-4665-1663-2 (eBook - PDF)

This book contains information obtained from authentic and highly regarded sources. Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Trademark Notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

Visit the Taylor & Francis Web site at
http://www.taylorandfrancis.com

and the CRC Press Web site at
http://www.crcpress.com

Dedicated to my graduate students, who taught me as much as
I taught them.

Contents

Series Preface

Half a century after their commercial introduction, composite materials are of widespread use in many industries. Applications such as aerospace, windmill blades, highway bridge retrofit, and many more require designs that assure safe and reliable operation for twenty years or more. Using composite materials, virtually any property, such as stiffness, strength, thermal conductivity, and fire resistance, can be tailored to the users needs by selecting the constituent material, their proportion and geometrical arrangement, and so on. In other words, the engineer is able to design the material concurrently with the structure. Also, modes of failure are much more complex in composites than in classical materials. Such demands for performance, safety, and reliability require that engineers consider a variety of phenomena during the design. Therefore, the aim of the *Composite Materials: Analysis and Design* book series is to bring to the design engineer a collection of works written by experts on every aspect of composite materials that is relevant to their design.

Variety and sophistication of material systems and processing techniques has grown exponentially in response to an ever-increasing number and type of applications. Given the variety of composite materials available as well as their continuous change and improvement, understanding of composite materials is by no means complete. Therefore, this book series serves not only the practicing engineer but also the researcher and student who are looking to advance the state-of-the-art in understanding material and structural response and developing new engineering tools for modeling and predicting such responses.

Thus, the series is focused on bringing to the public existing and developing knowledge about the material-property relationships, processing-property relationships, and structural response of composite materials and structures. The series scope includes analytical, experimental, and numerical methods that have a clear impact on the design of composite structures.

Ever Barbero, book series editor
West Virginia University, Morgantown, WV

Preface

Finite Element Analysis of Composite Materials deals with the analysis of structures made of composite materials, also called *composites.* The analysis of composites treated in this textbook includes the analysis of the material itself, at the micro-level, and the analysis of structures made of composite materials. This textbook evolved from the class notes of *MAE 646 Advanced Mechanics of Composite Materials* that I teach as a graduate course at West Virginia University. Although this is also a textbook on advanced mechanics of composite materials, the use of the finite element method is essential for the solution of the complex boundary value problems encountered in the advanced analysis of composites, and thus the title of the book.

There are a number of good textbooks on advanced mechanics of composite materials, but none carries the theory to a practical level by actually solving problems, as it is done in this textbook. Some books devoted exclusively to finite element analysis include some examples about modeling composites but fall quite short of dealing with the actual analysis and design issues of composite materials and composite structures. This textbook includes an explanation of the concepts involved in the detailed analysis of composites, a sound explanation of the mechanics needed to translate those concepts into a mathematical representation of the physical reality, and a detailed explanation of the solution of the resulting boundary value problems by using commercial Finite Element Analysis software such as *Abaqus*$^{\text{TM}}$. Furthermore, this textbook includes more than fifty fully developed examples interspersed with the theory, as well as more than seventy-five exercises at the end of chapters, and more than fifty separate pieces of *Abaqus* pseudocode used to explain in detail the solution of example problems. The reader will be able to reproduce the examples and complete the exercises. When a finite element analysis is called for, the reader will be able to do it with commercially or otherwise available software. A Web site is set up with links to download the necessary software unless it is easily available from Finite Element Analysis software vendors. Use of *Abaqus* and *MATLAB*$^{\text{TM}}$is explained with numerous examples, and the relevant code can be downloaded from the Web site. Furthermore, the reader will be able to extend the capabilities of *Abaqus* by use of user material subroutines and Python scripting, as demonstrated in the examples included in this textbook.

Chapters 1 through 7 can be covered in a one-semester graduate course. Chapter 2 (Introduction to Finite Element Analysis) contains a brief introduction intended for those readers who have not had a formal course or prior knowledge about the finite element method. Chapter 4 (Buckling) is not referenced in the remainder of the textbook and thus it could be omitted in favor of more exhaustive coverage of content in later chapters. Chapters 7 (Viscoelasticity), 8 (Continuum Damage Mechanics), and 9 (Discrete Damage Mechanics) are placed consecutively to emphasize hereditary phenomena. However, Chapter 7 can be skipped if more emphasis on damage and/or delaminations is desired in a one-semester course.

The *inductive method* is applied as much as possible in this textbook. That is, topics are introduced with examples of increasing complexity, until sufficient physical understanding is reached to introduce the general theory without difficulty. This method will sometimes require that, at earlier stages of the presentation, certain facts, models, and relationships be accepted as fact, until they are completely proven later on. For example, in Chapter 7, viscoelastic models are introduced early to aid the reader in gaining an appreciation for the response of viscoelastic materials. This is done simultaneously with a cursory introduction to the superposition principle and the Laplace transform, which are formally introduced only later in the chapter. For those readers accustomed to the *deductive method*, this may seem odd, but many years of teaching have convinced me that students acquire and retain knowledge more efficiently in this way.

It is assumed that the reader is familiar with basic mechanics of composites as covered in introductory level textbooks such as my previous textbook, *Introduction to Composite Material Design–Second Edition.* Furthermore, it is assumed that the reader masters a body of knowledge that is commonly acquired as part of a bachelor of science degree in any of the following disciplines: Aerospace, Mechanical, Civil, or similar. References to books and to other sections in this textbook, as well as footnotes, are used to assist the reader in refreshing those concepts and to clarify the notation used. Prior knowledge of continuum mechanics, tensor analysis, and the finite element method would enhance the learning experience but are not necessary for studying with this textbook. The finite element method is used as a tool to solve practical problems. For the most part, *Abaqus* is used throughout the book. Computing programming using Fortran, Python©, and *MATLAB* is limited to programming material models and post-processing algorithms. Basic knowledge of these programming languages is useful but not essential.

Only three software packages are used throughout the book. *Abaqus* is needed for finite element solution of numerous examples and suggested problems. *MATLAB* is needed for both symbolic and numerical solution of examples and suggested problems. Additionally, BMI3©, which is available free of charge on the book's Web site, is used in Chapter 4. Several other programs such as *ANSYS* Mechanical®, *LS-DYNA*®, *MSC-MARC*®, SolidWorks™ are cited, but not used in the examples. Relevant code used in the examples is available in the book's Web site `http://barbero.cadec-online.com/feacm-abaqus/`.

Composite materials are now ubiquitous in the marketplace, including extensive applications in aerospace, automotive, civil infrastructure, sporting goods, and so on. Their design is especially challenging because, unlike conventional materials such as metals, the composite material itself is designed concurrently with the composite structure. Preliminary design of composites is based on the assumption of a state of plane stress in the laminate. Furthermore, rough approximations are made about the geometry of the part, as well as the loading and support conditions. In this way, relatively simple analysis methods exist and computations can be carried out simply using algebra. However, preliminary analysis methods have a number of shortcomings that are remedied with advanced mechanics and finite element analysis, as explained in this textbook. Recent advances in commercial finite element analysis packages, with user-friendly pre- and post-processing, as well as powerful user-programmable features, have made detailed analysis of composites quite accessible to the designer. This textbook bridges the gap between powerful finite element tools and practical problems in structural analysis of composites. I expect that many graduate students, practicing engineers, and instructors will find this to be a useful and practical textbook on finite element analysis of composite materials based on sound understanding of advanced mechanics of composite materials.

Ever J. Barbero, 2013

Abaqus™ and SolidWorks™are registered trademarks of Dassault Systèmes. Abaqus is developed by SIMULIA, the Dassault Systèmes brand for Realistic Simulation. `www.simulia.com`.

ANSYS® is a registered trademark of ANSYS Inc. `www.ansys.com`

LS-DYNA® is a registered trademark of Livermore Software Technology Corporation. `www.lstc.com`.

MATLAB® is a registered trademark of The MathWorks, Inc. For product information, please contact: The MathWorks, Inc. 3, Apple Hill Drive, Natick, MA 01760-2098 USA; Tel: 508 647 7000; Fax: 508 647 7001; E-mail: info@mathworks.com; Web: www.mathworks.com

MSC-MARC® is a registered trademark of MSC Software. `www.mscsoftware.com`

Python© 2001–2010 by the Python Software Foundation. `python.org/psf`

Acknowledgments

I wish to thank Elio Sacco, Raimondo Luciano, and Fritz Campo for their contributions to Chapter 6 and to Tom Damiani, Joan Andreu Mayugo, and Xavier Martinez, who taught the course in 2004, 2006, and 2009, making many corrections and additions to the course notes on which this textbook is based. Also, thanks to Dr. Eduardo Sosa for helping set up the imperfection analysis in Chapter 4 and to Prof. Fabrizio Greco for reviewing Chapter 10. Acknowledgment is due also to those who reviewed the manuscript including Enrique Barbero, Guillermo Creus, Luis Godoy, Paolo Lonetti, Severino Marques, Pizhong Qiao, Timothy Norman, Sonia Sanchez, and Eduardo Sosa. Furthermore, recognition is due to those who helped me compile the solutions manual, including Hermann Alcazar, John Sandro Rivas, and Rajiv Dastane. Also, I wish to thank John Sandro Rivas for manually checking that the pseudo code of the examples translates into viable *Abaqus*/CAE Release 6.10 keystrokes and mouse clicks that accurately represent the problem at hand. Also, many thanks to my colleagues and students for their valuable suggestions and contributions to this textbook. Lastly, my gratitude goes to my wife, Ana Maria, and to my children, Margaret and Daniel, who gave up many opportunities to bond with their dad so that I might write this book.

List of Symbols

Symbols Related to Mechanics of Orthotropic Materials

$\boldsymbol{\epsilon}$	Strain tensor
ε_{ij}	Strain components in tensor notation
ϵ_α	Strain components in contracted notation
ϵ_α^e	Elastic strain
ϵ_α^p	Plastic strain
λ	Lame constant
ν	Poisson's ratio
ν_{12}	In-plane Poisson's ratio
ν_{23}, ν_{13}	Interlaminar Poisson's ratios
ν_{xy}	Apparent laminate Poisson's ratio x-y
$\boldsymbol{\sigma}$	Stress tensor
σ_{ij}	Stress components in tensor notation
σ_α	Stress components in contracted notation
$[a]$	Transformation matrix for vectors
e_i	Unit vector components in global coordinates
e'_i	Unit vector components in materials coordinates
f_i, f_{ij}	Tsai-Wu coefficients
k	Bulk modulus
l, m, n	Direction cosines
$\widetilde{u}(\varepsilon_{ij})$	Strain energy per unit volume
u_i	Displacement vector components
x_i	Global directions or axes
x'_i	Materials directions or axes
$\mathbf{C}$	Stiffness tensor
C_{ijkl}	Stiffness in index notation
$C_{\alpha,\beta}$	Stiffness in contracted notation
E	Young's modulus
E_1	Longitudinal modulus
E_2	Transverse modulus
E_2	Transverse-thickness modulus
E_x	Apparent laminate modulus in the global x-direction
$G = \mu$	Shear modulus
G_{12}	In-plane shear modulus
G_{23}, G_{13}	Interlaminar shear moduli

G_{xy}	Apparent laminate shear modulus x-y
I_{ij}	Second-order identity tensor
I_{ijkl}	Fourth-order identity tensor
Q'_{ij}	Lamina stiffness components in lamina coordinates
$[R]$	Reuter matrix
$\mathbf{S}$	Compliance tensor
S_{ijkl}	Compliance in index notation
$S_{\alpha,\beta}$	Compliance in contracted notation
$[T]$	Coordinate transformation matrix for stress
$[\overline{T}]$	Coordinate transformation matrix for strain

Symbols Related to Finite Element Analysis

$\underline{\underline{\partial}}$	Strain-displacement equations in matrix form
$\underline{\epsilon}$	Six-element array of strain components
$\theta_x, \theta_y, \theta_z$	Rotation angles following the right-hand rule (Figure 2.19)
$\underline{\sigma}$	Six-element array of stress components
ϕ_x, ϕ_y	Rotation angles used in plate and shell theory
$\underline{a}$	Nodal displacement array
u_j^e	Unknown parameters in the discretization
$\underline{\underline{B}}$	Strain-displacement matrix
$\underline{\underline{C}}$	Stiffness matrix
$\underline{\underline{K}}$	Assembled global stiffness matrix
$\underline{\underline{K}}^e$	Element stiffness matrix
$\underline{N}$	Interpolation function array
N_j^e	Interpolation functions in the discretization
$\underline{P}^e$	Element force array
$\underline{P}$	Assembled global force array

Symbols Related to Elasticity and Strength of Laminates

γ_{xy}^0	In-plane shear strain
γ_{4u}	Ultimate interlaminar shear strain in the 2-3 plane
γ_{5u}	Ultimate interlaminar shear strain in the 1-3 plane
γ_{6u}	Ultimate in-plane shear strain
$\epsilon_x^0, \epsilon_y^0$	In-plane strains
ϵ_{1t}	Ultimate longitudinal tensile strain
ϵ_{2t}	Ultimate transverse tensile strain
ϵ_{3t}	Ultimate transverse-thickness tensile strain
ϵ_{1c}	Ultimate longitudinal compressive strain
ϵ_{2c}	Ultimate transverse compressive strain
ϵ_{3c}	Ultimate transverse-thickness compressive strain
κ_x, κ_y	Bending curvatures
κ_{xy}	Twisting curvature

ϕ_x, ϕ_y	Rotations of the middle surface of the shell (Figure 2.19)
c_4, c_5, c_6	Tsai-Wu coupling coefficients
t_k	Lamina thickness
u_0, v_0, w_0	Displacements of the middle surface of the shell
z	Distance from the middle surface of the shell
A_{ij}	Components of the extensional stiffness matrix $[A]$
B_{ij}	Components of the bending-extension coupling matrix $[B]$
D_{ij}	Components of the bending stiffness matrix $[D]$
$[E_0]$	Extensional stiffness matrix $[A]$, in ANSYS notation
$[E_1]$	Bending-extension matrix $[B]$, in ANSYS notation
$[E_2]$	Bending stiffness matrix $[D]$, in ANSYS notation
F_{1t}	Longitudinal tensile strength
F_{2t}	Transverse tensile strength
F_{3t}	Transverse-thickness tensile strength
F_{1c}	Longitudinal compressive strength
F_{2c}	Transverse compressive strength
F_{3c}	Transverse-thickness compressive strength
F_4	Interlaminar shear strength in the 2-3 plane
F_5	Interlaminar shear strength in the 1-3 plane
F_6	In-plane shear strength
H_{ij}	Components of the interlaminar shear matrix $[H]$
I_F	Failure index
M_x, M_y, M_xy	Moments per unit length (Figure 3.3)
$\widehat{M}_n$	Applied bending moment per unit length
N_x, N_y, N_xy	In-plane forces per unit length (Figure 3.3)
$\widehat{N}_n$	Applied in-plane force per unit length, normal to the edge
$\widehat{N}_{ns}$	Applied in-plane shear force per unit length, tangential
$\left(\overline{Q}_{ij}\right)_k$	Lamina stiffness components in laminate coordinates, layer k
V_x, V_y	Shear forces per unit length (Figure 3.3)

Symbols Related to Buckling

λ, λ_i	Eigenvalues
s	Perturbation parameter
Λ	Load multiplier
$\Lambda^{(cr)}$	Bifurcation multiplier or critical load multiplier
$\Lambda^{(1)}$	Slope of the post-critical path
$\Lambda^{(2)}$	Curvature of the post-critical path
v	Eigenvectors (buckling modes)
$[K]$	Stiffness matrix
$[K_s]$	Stress stiffness matrix
P_{CR}	Critical load

Symbols Related to Free Edge Stresses

$\eta_{xy,x}, \eta_{xy,y}$	Coefficients of mutual influence
$\eta_{x,xy}, \eta_{y,xy}$	Alternate coefficients of mutual influence
F_{yz}	Interlaminar shear force y-z
F_{xz}	Interlaminar shear force x-z
M_z	Interlaminar moment

Symbols Related to Micromechanics

$\overline{\epsilon}_\alpha$	Average engineering strain components
$\overline{\varepsilon}_{ij}$	Average tensor strain components
$\epsilon^0_\alpha, \varepsilon^0_{ij}$	Far-field applied strain components
$\overline{\sigma}_\alpha$	Average stress components
$\mathbf{A}^i$	Strain concentration tensor, i-th phase, contracted notation
$2a_1, 2a_2, 2a_3$	Dimensions of the RVE
A_{ijkl}	Components of the strain concentration tensor
$\mathbf{B}^i$	Stress concentration tensor, i-th phase, contracted notation
B_{ijkl}	Components of the stress concentration tensor
I	6×6 identity matrix
P_{ijkl}	Eshelby tensor
V_f	Fiber volume fraction
V_m	Matrix volume fraction

Symbols Related to Viscoelasticity

$\dot{\varepsilon}$	Stress rate
η	Viscosity
θ	Age or aging time
$\dot{\sigma}$	Stress rate
τ	Time constant of the material or system
Γ	Gamma function
s	Laplace variable
t	Time
$C_{\alpha,\beta}(t)$	Stiffness tensor in the time domain
$C_{\alpha,\beta}(s)$	Stiffness tensor in the Laplace domain
$\widehat{C}_{\alpha,\beta}(s)$	Stiffness tensor in the Carson domain
$D(t)$	Compliance
$D_0, (D_i)_0$	Initial compliance values
$D_c(t)$	Creep component of the total compliance $D(t)$
D', D''	Storage and loss compliances
$E_0, (E_i)_0$	Initial moduli
E_∞	Equilibrium modulus
E, E_0, E_1, E_2	Parameters in the viscoelastic models (Figure 7.1)
$E(t)$	Relaxation

E', E''	Storage and loss moduli
$F[]$	Fourier transform
$(G_{ij})_0$	Initial shear moduli
$H(t - t_0)$	Heaviside step function
$H(\theta)$	Relaxation spectrum
$L[]$	Laplace transform
$L[]^{-1}$	Inverse Laplace transform

Symbols Related to Damage

α	Laminate CTE
$\alpha^{(k)}$	CTE of lamina k
α_{cr}	Critical misalignment angle at longitudinal compression failure
α_σ	Standard deviation of fiber misalignment
$\gamma(\delta)$	Damage hardening function
γ_0	Damage threshold
δ_{ij}	Kronecker delta
δ	Damage hardening variable
ε	Effective strain
$\overline{\varepsilon}$	Undamaged strain
ε^p	Plastic strain
$\dot{\gamma}$	Heat dissipation rate per unit volume
$\dot{\gamma}_s$	Internal entropy production rate
λ	Crack density
λ_{lim}	Saturation crack density
$\dot{\lambda}, \dot{\lambda}^d$	Damage multiplier
$\dot{\lambda}^p$	Yield multiplier
ρ	Density
σ	Effective stress
$\overline{\sigma}$	Undamaged stress
τ_{13}, τ_{23}	Intralaminar shear stress components
φ, φ^*	Strain energy density, and complementary SED
χ	Gibbs energy density
ψ	Helmholtz free energy density
ΔT	Change in temperature
$\mathbf{\Omega} = \Omega_{ij}$	Integrity tensor
$2a_0$	Representative crack size
d_i	Eigenvalues of the damage tensor
f^d	Damage flow surface
f^p	Yield flow surface
$f(x), F(x)$	Probability density, and its cumulative probability
g	Damage activation function
g^d	Damage surface
g^p	Yield surface
h	Laminate thickness

h_k	Thickness of lamina k
m	Weibull modulus
p	Yield hardening variable
$\widehat{p}$	Thickness average of quantity p
$\widetilde{p}$	Virgin value of quantity p
$\overline{p}$	Volume average of quantity p
$\mathbf{q}$	Hear flow vector per unit area
r	Radiation heat per unit mass
s	Specific entropy
$u(\varepsilon_{ij})$	Internal energy density
A	Crack area
$[A]$	Laminate in-plane stiffness matrix
A_{ijkl}	Tension-compression damage constitutive tensor
B_{ijkl}	Shear damage constitutive tensor
B_a	Dimensionless number (8.57)
$\overline{C}_{\alpha,\beta}$	Stiffness matrix in the undamaged configuration
$\mathbf{C}^{ed}$	Tangent stiffness tensor
D_{ij}	Damage tensor
D_{1t}^{cr}	Critical damage at longitudinal tensile failure
D_{1c}^{cr}	Critical damage at longitudinal compression failure
D_{2t}^{cr}	Critical damage at transverse tensile failure
D_2, D_6	Damage variables
$E(D)$	Effective modulus
$\overline{E}$	Undamaged (virgin) modulus
$G_c = 2\gamma_c$	Surface energy
G_{Ic}, G_{IIc}	Critical energy release rate in modes I and II
J_{ijkl}	Normal damage constitutive tensor
M_{ijkl}	Damage effect tensor
N	Number of laminas in the laminate
$\{N\}$	Membrane stress resultant array
Q	Degraded 3x3 stiffness matrix of the laminate
$R(p)$	Yield hardening function
R_0	Yield threshold
S	Entropy or Laminate complinace matrix, depending on context
T	Temperature
U	Strain energy
V	Volume of the RVE
Y_{ij}	Thermodynamic force tensor

Symbols Related to Delaminations

α	Mixed mode crack propagation exponent
β_δ, β_G	Mixed mode ratios
δ	CZM separation of the interface
δ_m	Mixed mode separation

δ_m^0	Mixed mode separation at damage onset
δ_m^0	Mixed mode separation at fracture
σ^0	CZM critical separation at damage onset
ℓ	Delamination length for 2D delaminations
σ^0	CZM strength of the interface
ψ_{xi}, ψ_{yi}	Rotation of normals to the middle surface of the plate
Ω	Volume of the body
Ω_D	Delaminated region
Π_e	Potential energy, elastic
Π^r	Potential energy, total
$\dot{\Gamma}$	Dissipation rate
Λ	Interface strain energy density per unit area
$\partial\Omega$	Boundary of the body
d	One-dimensional damage state variable
k_{xy}, k_z	Displacement continuity parameters
$[A_i], [B_i], [D_i]$	Laminate stiffness sub-matrices
D_I, D_{II}, D_{III}	Damage variables for modes I, II, and III of CZM
$G(\ell)$	Energy release rate (ERR), total, in 2D
G	Energy release rate (ERR), total, in 3D
G_I, G_{II}, G_{III}	Energy release rate (ERR) of modes I, II, and III
G_c	Critical energy release rate (ERR), total, in 3D
G_I^c	Critical energy release rate mode I
$[H_i]$	Laminate interlaminar shear stiffness matrix
K	Penalty stiffness
$\tilde{K}$	Virgin penalty stiffness
K_I, K_{II}, K_{III}	Stress intensity factors (SIF) of modes I, II, and III
N_i, M_i, T_i	Stress resultants
U	Internal energy
W	Work done by the body on its surroundings
$W_{closure}$	Crack closure work

List of Examples

Chapter 1

Mechanics of Orthotropic Materials

This chapter provides the foundation for the rest of the book. Basic concepts of mechanics, tailored for composite materials, are presented, including coordinate transformations, constitutive equations, and so on. Continuum mechanics is used to describe deformation and stress in an orthotropic material. The basic equations are reviewed in Sections 1.2 to 1.9. Tensor operations are reviewed in Section 1.10 because they are used in the rest of the chapter. Coordinate transformations are required to express quantities such as stress, strain, and stiffness in lamina coordinates, in laminate coordinates, and so on. They are reviewed in Sections 1.10 to 1.11. This chapter is heavily referenced in the rest of the book, and thus readers who are already versed in continuum mechanics may choose to come back to review this material as needed.

1.1 Lamina Coordinate System

A single lamina of fiber reinforced composite behaves as an orthotropic material. That is, the material has three mutually perpendicular planes of symmetry. The intersection of these three planes defines three axes that coincide with the fiber direction (x'_1), the thickness coordinate (x'_3), and a third direction $x'_2 = x'_3 \times x'_1$ perpendicular to the other two[1] [1].

1.2 Displacements

Under the action of forces, every point in a body may translate and rotate as a rigid body as well as deform to occupy a new region. The displacements u_i of any point P in the body (Figure 1.1) are defined in terms of the three components of the vector u_i (in a rectangular Cartesian coordinate system) as $u_i = (u_1,u_2,u_3)$. An

[1] × denotes vector cross product.

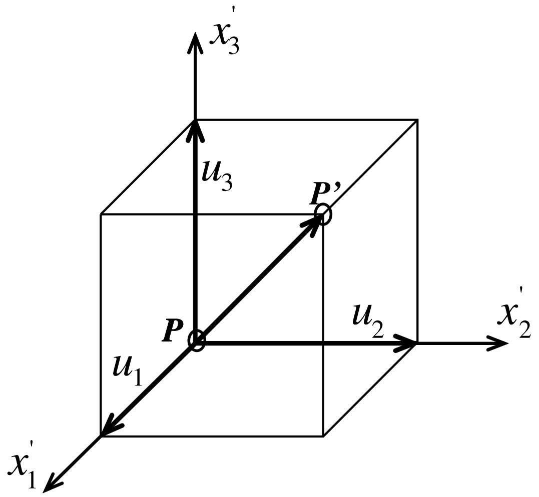


Fig. 1.1: Notation for displacement components.

alternate notation for displacements is $u_i = (u, v, w)$. Displacement is a vector or first-order tensor quantity

$$\mathbf{u} = u_i = (u_{1,} u_{2,} u_3) \ ; \ i = 1...3 \tag{1.1}$$

where boldface (e.g., $\mathbf{u}$) indicates a *tensor* written in tensor notation, in this case a vector (or first-order tensor). In this book, all tensors are boldfaced (e.g., $\boldsymbol{\sigma}$), but their components are not (e.g., σ_{ij}). The order of the tensor (i.e., first, second, fourth, etc.) must be inferred from context, or as in (1.1), by looking at the number of subscripts of the same entity written in index notation (e.g., u_i).

1.3 Strain

For geometric nonlinear analysis, the components of the Lagrangian strain tensor are [2]

$$L_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i} + u_{r,i}u_{r,j}) \tag{1.2}$$

where

$$u_{i,j} = \frac{\partial u_i}{\partial x_j} \tag{1.3}$$

If the gradients of the displacements are so small that products of partial derivatives of u_i are negligible compared with linear (first-order) derivative terms, then the (infinitesimal) strain tensor ε_{ij} is given by [2]

$$\boldsymbol{\varepsilon} = \varepsilon_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i}) \tag{1.4}$$

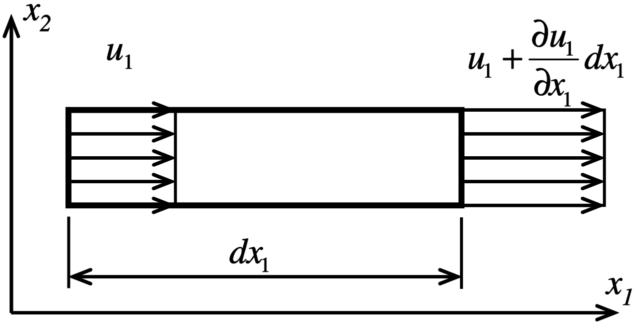


Fig. 1.2: Normal strain.

Again, boldface indicates a tensor, the order of which is implied from the context. For example ε is a one-dimensional strain and $\boldsymbol{\varepsilon}$ is the second-order tensor of strain. Index notation (e.g., $= \varepsilon_{ij}$) is used most of the time and the tensor character of variables (scalar, vector, second order, and so on) is easily understood from context.

From the definition (1.4), strain is a second-order, symmetric tensor (i.e., $\varepsilon_{ij} = \varepsilon_{ji}$). In expanded form the strains are defined by

$$\begin{aligned}
\varepsilon_{11} &= \frac{\partial u_1}{\partial x_1} = \epsilon_1 \; ; \quad 2\varepsilon_{12} = 2\varepsilon_{21} = \left(\frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1}\right) = \gamma_6 = \epsilon_6 \\
\varepsilon_{22} &= \frac{\partial u_2}{\partial x_2} = \epsilon_2 \; ; \quad 2\varepsilon_{13} = 2\varepsilon_{31} = \left(\frac{\partial u_1}{\partial x_3} + \frac{\partial u_3}{\partial x_1}\right) = \gamma_5 = \epsilon_5 \\
\varepsilon_{33} &= \frac{\partial u_3}{\partial x_3} = \epsilon_3 \; ; \quad 2\varepsilon_{23} = 2\varepsilon_{32} = \left(\frac{\partial u_2}{\partial x_3} + \frac{\partial u_3}{\partial x_2}\right) = \gamma_4 = \epsilon_4
\end{aligned} \tag{1.5}$$

where ϵ_α with $\alpha = 1..6$ are defined in Section 1.5. The normal components of strain ($i = j$) represent the change in length per unit length (Figure 1.2). The shear components of strain ($i \neq j$) represent one-half the change in an original right angle (Figure 1.3). The engineering shear strain $\gamma_\alpha = 2\varepsilon_{ij}$, for $i \neq j$, is often used instead of the tensor shear strain because the shear modulus G is defined by $\tau = G\gamma$ in mechanics of materials [3]. The strain tensor, being of second order, can be displayed as a matrix

$$[\varepsilon] = \begin{bmatrix} \varepsilon_{11} & \varepsilon_{12} & \varepsilon_{13} \\ \varepsilon_{12} & \varepsilon_{22} & \varepsilon_{23} \\ \varepsilon_{13} & \varepsilon_{23} & \varepsilon_{33} \end{bmatrix} = \begin{bmatrix} \epsilon_1 & \epsilon_6/2 & \epsilon_5/2 \\ \epsilon_6/2 & \epsilon_2 & \epsilon_4/2 \\ \epsilon_5/2 & \epsilon_4/2 & \epsilon_3 \end{bmatrix} \tag{1.6}$$

where [ ] is used to denote matrices.

1.4 Stress

The stress vector associated to a plane passing through a point is the force per unit area acting on the plane passing through the point. A second-order tensor, called stress tensor, completely describes the state of stress at a point. The stress tensor

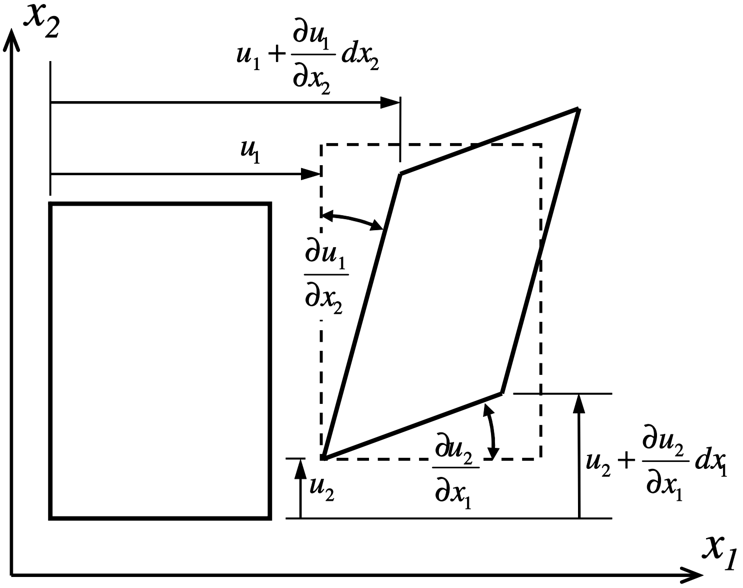

x_2
$u_1 + \frac{\partial u_1}{\partial x_2} dx_2$
u_1
$\frac{\partial u_1}{\partial x_2}$
$\frac{\partial u_2}{\partial x_1}$
u_2
$u_2 + \frac{\partial u_2}{\partial x_1} dx_1$
x_1

Fig. 1.3: Engineering shear strain.

can be expressed in terms of the components acting on three mutually perpendicular planes aligned with the orthogonal coordinate directions as indicated in Figure 1.4. The tensor notation for stress is σ_{ij} with $(i, j = 1, 2, 3)$, where the first subscript corresponds to the direction of the normal to the plane of interest and the second subscript corresponds to the direction of the stress. Tensile normal stresses $(i = j)$ are defined to be positive when the normal to the plane and the stress component directions are either both positive or both negative. All components of stress depicted in Figure 1.4 have a positive sense. Force and moment equilibrium of the element in Figure 1.4 requires that the stress tensor be symmetric (i.e., $\sigma_{ij} = \sigma_{ji}$) [3]. The stress tensor, being of second order, can be displayed as a matrix

$$[\sigma] = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{12} & \sigma_{22} & \sigma_{23} \\ \sigma_{13} & \sigma_{23} & \sigma_{33} \end{bmatrix} = \begin{bmatrix} \sigma_1 & \sigma_6 & \sigma_5 \\ \sigma_6 & \sigma_2 & \sigma_4 \\ \sigma_5 & \sigma_4 & \sigma_3 \end{bmatrix} \tag{1.7}$$

1.5 Contracted Notation

Since the stress is symmetric, it can be written in *Voigt* contracted notation as

$$\sigma_\alpha = \sigma_{ij} = \sigma_{ji} \tag{1.8}$$

with the contraction rule defined as follows

$$\begin{aligned} \alpha &= i && if \quad i = j \\ \alpha &= 9 - i - j && if \quad i \neq j \end{aligned} \tag{1.9}$$

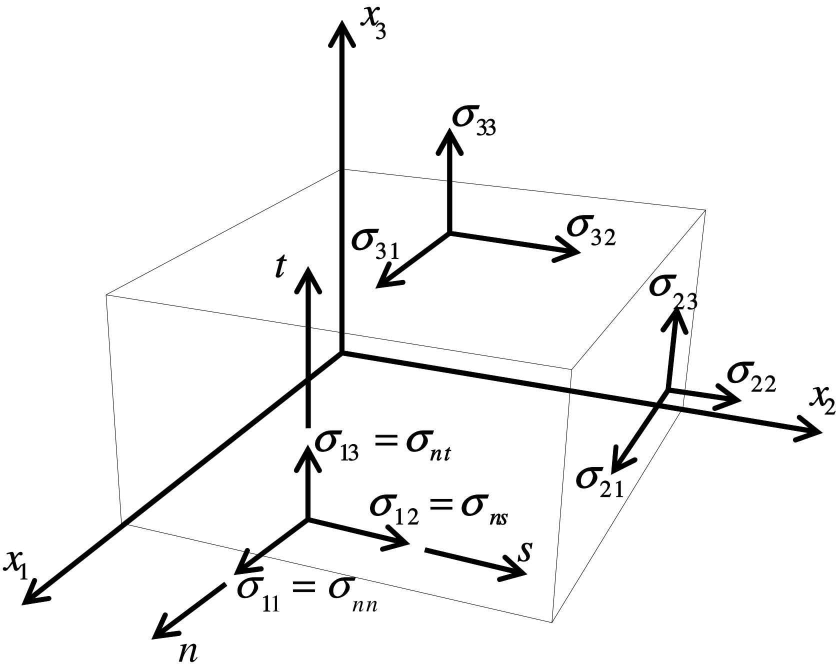


Fig. 1.4: Stress components.

Table 1.1: Contracted notation convention used by various FEA software packages.

Standard Convention	Abaqus/Standard	LS-DYNA and Abaqus/Explicit	ANSYS/Mechanical
$11 \longrightarrow 1$	$11 \longrightarrow 1$	$11 \longrightarrow 1$	$11 \longrightarrow 1$
$22 \longrightarrow 2$	$22 \longrightarrow 2$	$22 \longrightarrow 2$	$22 \longrightarrow 2$
$33 \longrightarrow 3$	$33 \longrightarrow 3$	$33 \longrightarrow 3$	$33 \longrightarrow 3$
$23 \longrightarrow 4$	$12 \longrightarrow 4$	$12 \longrightarrow 4$	$12 \longrightarrow 4$
$13 \longrightarrow 5$	$13 \longrightarrow 5$	$23 \longrightarrow 5$	$23 \longrightarrow 5$
$12 \longrightarrow 6$	$23 \longrightarrow 6$	$13 \longrightarrow 6$	$13 \longrightarrow 6$

resulting in the contracted version of stress components shown in (1.7). The same applies to the strain tensor, resulting in the contracted version of strain shown in (1.6). Note that the six components of stress σ_α with $\alpha = 1 \dots 6$ can be arranged into a column array, denoted by curly brackets $\{\ \}$ as in (1.10), but $\{\sigma\}$ is not a vector, but just a convenient way to arrange the six unique components of a symmetric second-order tensor.

1.5.1 Alternate Contracted Notation

Some FEA software packages use different contracted notations, as shown in Table 1.1. For example, to transform stresses or strains from standard notation to *Abaqus* notation, a transformation matrix can be used as follows

$$\{\sigma_A\} = [T]\{\sigma\} \tag{1.10}$$

where the subscript $()_A$ denotes a quantity in ABAQUS notation. Also note that { } denotes a column array, in this case of six elements, and [ ] denotes a matrix, in this case the 6 × 6 rotation matrix given by

$$[T] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{1.11}$$

The stiffness matrix transforms as follows

$$[C_A] = [T]^T[C][T] \tag{1.12}$$

For *LS-DYNA*and *ANSYS*, the transformation matrix is

$$[T] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{1.13}$$

1.6 Equilibrium and Virtual Work

The three equations of equilibrium at every point in a body are written in tensor notation as

$$\sigma_{ij,j} + f_i = 0 \tag{1.14}$$

where f_i is the body force per unit volume and $(\ )_{,j} = \dfrac{\partial}{\partial x_j}$. When body forces are negligible, the expanded form of the equilibrium equations, written in the laminate coordinate system x-y-z, is

$$\begin{aligned} \frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + \frac{\partial \sigma_{xz}}{\partial z} &= 0 \\ \frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + \frac{\partial \sigma_{yz}}{\partial z} &= 0 \\ \frac{\partial \sigma_{xz}}{\partial x} + \frac{\partial \sigma_{yz}}{\partial y} + \frac{\partial \sigma_{zz}}{\partial z} &= 0 \end{aligned} \tag{1.15}$$

The principle of virtual work (PVW) provides an alternative to the equations of equilibrium [4]. Since the PVW is an integral expression, it is more convenient than (1.14) for finite element formulation. The PVW reads

$$\int_V \sigma_{ij}\delta\epsilon_{ij}dV - \int_S t_i\delta u_i dS - \int_V f_i\delta u_i dV = 0 \tag{1.16}$$

where t_i are the surface tractions per unit area acting on the surface S. The negative sign means that work is done by external forces (t_i, f_i) on the body. The forces and the displacements follow the same sign convention; that is, a component is positive when it points in the positive direction of the respective axis. The first term in (1.16) is the virtual work performed by the internal stresses and it is positive following the same sign convention.

Example 1.1 *Find the displacement function $u(x)$ for a slender rod of cross-sectional area A, length L, modulus E, and density ρ, hanging from the top end and subjected to its own weight. Use a coordinate x pointing downward with origin at the top end.*

Solution to Example 1.1 *We assume a quadratic displacement function*

$$u(x) = C_0 + C_1x + C_2x^2$$

Using the boundary condition (B.C.) at the top yields $C_0 = 0$. The PVW (1.16) simplifies because the only non-zero strain is ϵ_x and there is no surface tractions. Using Hooke's law

$$\int_0^L E\epsilon_x\delta\epsilon_x A dx - \int_0^L \rho g\delta u A dx = 0$$

From the assumed displacement

$$\begin{aligned}\delta u &= x\delta C_1 + x^2\delta C_2 \\ \epsilon_x &= \frac{du}{dx} = C_1 + 2xC_2 \\ \delta\epsilon_x &= \delta C_1 + 2x\delta C_2\end{aligned}$$

Substituting

$$EA\int_o^L (C_1 + 2xC_2)(\delta C_1 + 2x\delta C_2)dx - \rho g A\int_0^L (x\delta C_1 + x^2\delta C_2)dx = 0$$

Integrating and collecting terms in δC_1 and δC_2 separately

$$(EC_2L^2 + EC_1L - \frac{\rho g L^2}{2})\delta C_1 + (\frac{4}{3}EC_2L^3 + EC_1L^2 - \frac{\rho g L^3}{3})\delta C_2 = 0$$

Since δC_1 and δC_2 have arbitrary (virtual) values, two equations in two unknowns are obtained, one inside each parenthesis. Solving them we get

$$C_1 = \frac{L\rho g}{E} \ ; C_2 = -\frac{\rho g}{2E}$$

Substituting back into $u(x)$

$$u(x) = \frac{\rho g}{2E}(2L - x)x$$

which coincides with the exact solution from mechanics of materials.

1.7 Boundary Conditions

1.7.1 Traction Boundary Conditions

The solution of problems in solid mechanics requires that boundary conditions be specified. The boundary conditions may be specified in terms of components of displacement, stress, or a combination of both. For any point on an arbitrary surface, the traction T_i is defined as the vector consisting of the three components of stress acting on the surface at the point of interest. As indicated in Figure 1.4 the traction vector consists of one component of normal stress, σ_{nn}, and two components of shear stress, σ_{nt} and σ_{ns}. The traction vector can be written using Cauchy's law

$$T_i = \sigma_{ji} n_j = \sum_j^3 \sigma_{ji} n_j \tag{1.17}$$

where n_j is the unit normal to the surface at the point under consideration.[2] For a plane perpendicular to the x_1 axis $n_i = (1, 0, 0)$ and the components of the traction are $T_1 = \sigma_{11}$, $T_2 = \sigma_{12}$, and $T_3 = \sigma_{13}$.

1.7.2 Free Surface Boundary Conditions

The condition that a surface be free of stress is equivalent to all components of traction being zero, i.e., $T_n = \sigma_{nn} = 0$, $T_t = \sigma_{nt} = 0$, and $T_s = \sigma_{ns} = 0$. It is possible that only selected components of the traction be zero while others are non-zero. For example, pure pressure loading corresponds to non-zero normal stress and zero shear stresses.

1.8 Continuity Conditions

1.8.1 Traction Continuity

Equilibrium (action and reaction) requires that the traction components T_i must be continuous across any surface. Mathematically this is stated as $T_i^+ - T_i^- = 0$. Using (1.17), $T_i^+ = \sigma_{ji}^+ n_j$. Since $n_j^+ = -n_j^-$, we have $\sigma_{ji}^+ = \sigma_{ji}^-$. In terms of individual stress components, $\sigma_{nn}^+ = \sigma_{nn}^-$, $\sigma_{nt}^+ = \sigma_{nt}^-$, and $\sigma_{ns}^+ = \sigma_{ns}^-$ (Figure 1.5). Thus, the normal and shear components of stress acting on a surface must be continuous across that surface. There are no continuity requirements on the other three components of stress. That is, it is possible that $\sigma_{tt}^+ \neq \sigma_{tt}^-$, $\sigma_{ss}^+ \neq \sigma_{ss}^-$, and $\sigma_{ts}^+ \neq \sigma_{ts}^-$. Lack of continuity of the two normal and one shear components of stress is very common because the material properties are discontinuous across layer boundaries.

[2] Einstein's summation convention can be introduced with (1.17) as an example. Any pair of repeated indices implies a summation over all the values of the index in question. Furthermore, each pair of repeated indices represents a *contraction*. That is, the order of resulting tensor, in this case order one for T_i, is two less than the sum of the orders of the tensors involved in the operation. The resulting tensor keeps only the *free* indices that are not involved in the contraction–in this case only i remains.

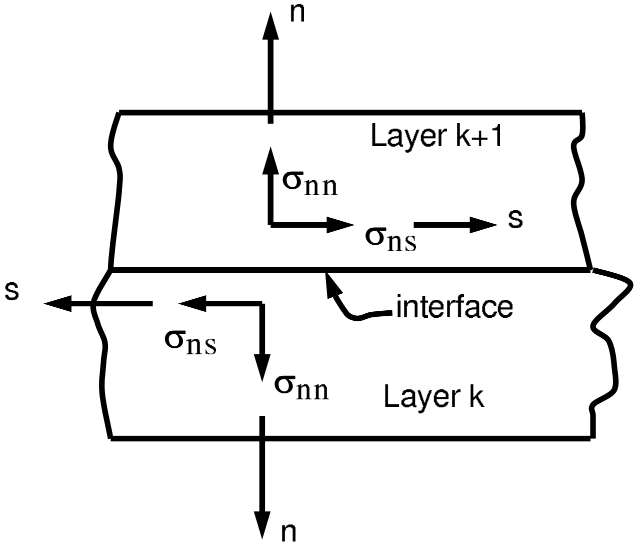


Fig. 1.5: Traction continuity across an interface.

1.8.2 Displacement Continuity

Certain conditions on displacements must be satisfied along any surface in a perfectly bonded continuum. Consider for example buckling of a cylinder under external pressure (Figure 1.6). The displacements associated with the material from either side of the line A-A must be identical $u_i^+ = u_i^-$. The continuity conditions must be satisfied at every point in a perfectly bonded continuum. However, continuity is not required in the presence of de-bonding or sliding between regions or phases of a material. For the example shown, continuity of slope $(\frac{\partial w^+}{\partial \theta} = \frac{\partial w^-}{\partial \theta})$, must be satisfied, where w is the radial displacement.

1.9 Compatibility

The strain displacement equations (1.5) provide six equations for only three unknown displacements u_i. Thus, integration of equations (1.5) to determine the unknown displacements will not have a single-valued solution unless the strains ε_{ij} satisfy certain conditions. Arbitrary specification of the ε_{ij} could result in discontinuities in the material, including gaps and/or overlapping regions.

The necessary conditions for single-valued displacements are the *compatibility conditions.* Although these six equations are available [2], they are not used here because the displacement method, which is used throughout this book, does not require them. That is, in solving problems, the form of displacements u_i is always assumed a priori. Then, the strains are computed with (1.5), and the stress with (1.46). Finally, equilibrium is enforced by using the PVW (1.16).

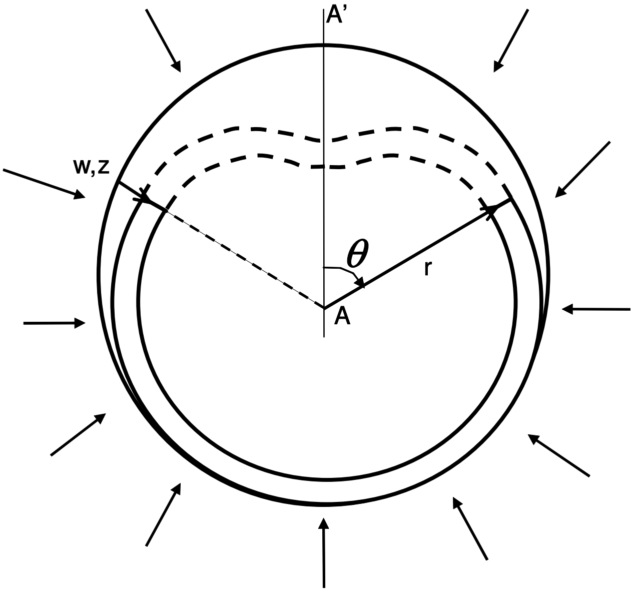


Fig. 1.6: Buckling of an encased cylindrical pipe under external pressure.

1.10 Coordinate Transformations

The coordinates of point P in the prime coordinate system can be found from its coordinates in the unprimed system. From Figure 1.7, the coordinates of point P are

$$\begin{aligned} x_1' &= x_1 \cos\theta + x_2 \sin\theta \\ x_2' &= -x_1 \sin\theta + x_2 \cos\theta \\ x_3' &= x_3 \end{aligned} \tag{1.18}$$

or

$$x_i' = a_{ij} x_j \tag{1.19}$$

or in matrix notation

$$\{x'\} = [a]\{x\} \tag{1.20}$$

where a_{ij} are the components of the unit vectors of the primed system e_i' on the unprimed system e_j, by rows [2]

$$a_{ij} = \cos(e_i', e_j) = \begin{array}{|c|c|c|c|} \hline & e_1 & e_2 & e_3 \\ \hline e_1' & a_{11} & a_{12} & a_{13} \\ \hline e_2' & a_{21} & a_{22} & a_{23} \\ \hline e_3' & a_{31} & a_{32} & a_{33} \\ \hline \end{array} \tag{1.21}$$

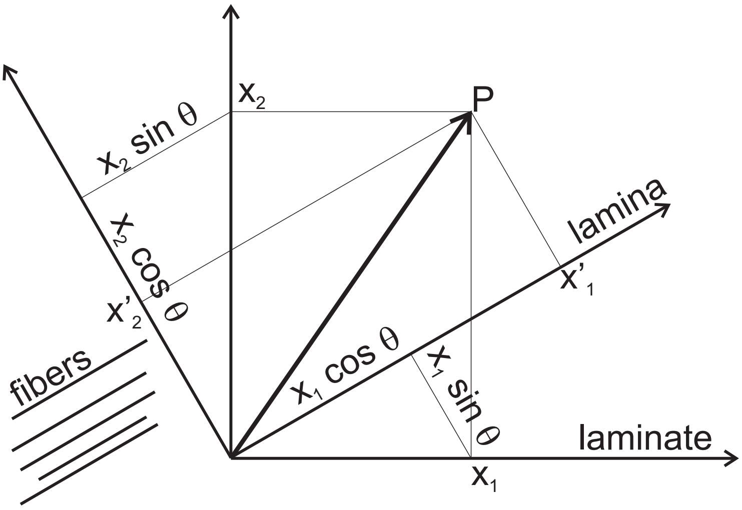


Fig. 1.7: Coordinate transformation.

If primed coordinates denote the lamina coordinates and unprimed denote the laminate coordinates, then (1.19) transforms vectors from laminate to lamina coordinates. The inverse transformation simply uses the transpose matrix

$$\{x\} = [a]^T \{x'\} \tag{1.22}$$

Example 1.2 *A composite layer has fiber orientation $\theta = 30°$. Construct the $[a]$ matrix by calculating the direction cosines of the lamina system, i.e., the components of the unit vectors of the lamina system (x'_i) on the laminate system (x_j).*

Solution to Example 1.2 *From Figure 1.7 and (1.19) we have*

$$\begin{aligned}
a_{11} &= \cos\theta = \frac{\sqrt{3}}{2} \\
a_{12} &= \sin\theta = \frac{1}{2} \\
a_{13} &= 0 \\
a_{21} &= -\sin\theta = -\frac{1}{2} \\
a_{22} &= \cos\theta = \frac{\sqrt{3}}{2} \\
a_{23} &= 0 \\
a_{31} &= 0 \\
a_{32} &= 0 \\
a_{33} &= 1
\end{aligned}$$

Example 1.3 *A fiber reinforced composite tube is wound in the hoop direction (1-direction). Formulas for the stiffness values (E_1, E_2, etc.) are given in that system. However, when analyzing the cross-section of this material with generalized plane strain elements (CAX4 in*

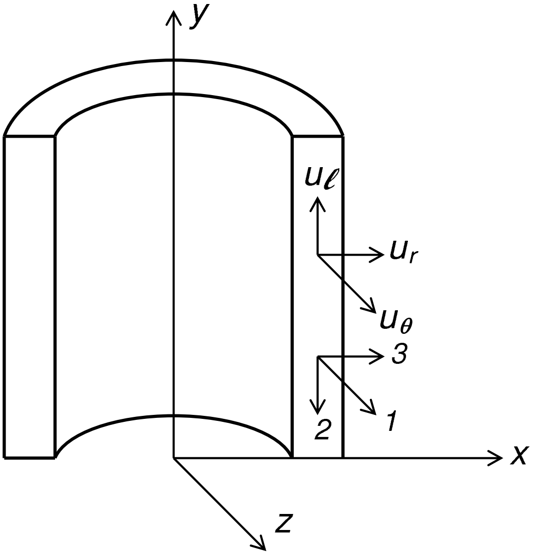


Fig. 1.8: Coordinate transformation for axial-symmetric analysis.

Abaqus*), the model is typically constructed in the structural X, Y, Z system. It is therefore necessary to provide the stiffness values in the structural system as E_x, E_y, etc. Construct the transformation matrix $[a]^T$ to go from lamina coordinates (1-2-3) to structural coordinates in Figure 1.8.*

Solution to Example 1.3 *First, construct $[a]$ using the definition (1.21). Taking each unit vector (1-2-3) at a time we construct the matrix $[a]$ by rows. The i-th row contains the components of (i=1,2,3) along (X-Y-Z).*

$[a]$	X	Y	Z
1	0	0	1
2	0	-1	0
3	1	0	0

The required transformation is just the transpose of the matrix above.

1.10.1 Stress Transformation

A second-order tensor σ_{pq} can be thought as the (un-contracted) outer product[3] of two vectors V_p and V_q

$$\sigma_{pq} = V_p \otimes V_q \tag{1.23}$$

each of which transforms as (1.19)

$$\sigma'_{ij} = a_{ip} V_p \otimes a_{jq} V_q \tag{1.24}$$

Therefore,

[3] The outer product preserves all indices of the entities involved, thus creating a tensor of order equal to the sum of the order of the entities involved.

$$\sigma'_{ij} = a_{ip} a_{jq} \sigma_{pq} \tag{1.25}$$

or, in matrix notation

$$\{\sigma'\} = [a]\{\sigma\}[a]^T \tag{1.26}$$

For example, expand σ'_{11} in contracted notation

$$\sigma'_1 = a_{11}^2 \sigma_1 + a_{12}^2 \sigma_2 + a_{13}^2 \sigma_3 + 2a_{11}a_{12}\sigma_6 + 2a_{11}a_{13}\sigma_5 + 2a_{12}a_{13}\sigma_4 \tag{1.27}$$

Expanding σ'_{12} in contracted notation yields

$$\begin{aligned} \sigma'_6 = {} & a_{11}a_{21}\sigma_1 + a_{12}a_{22}\sigma_2 + a_{13}a_{23}\sigma_3 + (a_{11}a_{22} + a_{12}a_{21})\sigma_6 \\ & + (a_{11}a_{23} + a_{13}a_{21})\sigma_5 + (a_{12}a_{23} + a_{13}a_{22})\sigma_4 \end{aligned} \tag{1.28}$$

The following algorithm is used to obtain a 6 × 6 coordinate transformation matrix [T] such that (1.25) is rewritten in contracted notation as

$$\sigma'_\alpha = T_{\alpha\beta} \sigma_\beta \tag{1.29}$$

If $\alpha \leq 3$ and $\beta \leq 3$ then $i = j$ and $p = q$, so

$$T_{\alpha\beta} = a_{ip} a_{ip} = a_{ip}^2 \quad \text{no sum on } i, p \tag{1.30}$$

If $\alpha \leq 3$ and $\beta > 3$ then $i = j$ but $p \neq q$, and taking into account that switching p by q yields the same value of $\beta = 9 - p - q$ as per (1.9) we have

$$T_{\alpha\beta} = a_{ip} a_{iq} + a_{iq} a_{ip} = 2 a_{ip} a_{iq} \quad \text{no sum on } i, p \tag{1.31}$$

If $\alpha > 3$, then $i \neq j$, but we want only one stress, say σ_{ij}, not σ_{ji} because they are numerically equal. In fact $\sigma_\alpha = \sigma_{ij} = \sigma_{ji}$ with $\alpha = 9 - i - j$. If in addition $\beta \leq 3$ then $p = q$ and we get

$$T_{\alpha\beta} = a_{ip} a_{jp} \quad \text{no sum on } i, p \tag{1.32}$$

When $\alpha > 3$ and $\beta > 3$, $i \neq j$ and $p \neq q$ so we get

$$T_{\alpha\beta} = a_{ip} a_{jq} + a_{iq} a_{jp} \tag{1.33}$$

which completes the derivation of $T_{\alpha\beta}$. Expanding (1.30–1.33) and using (1.21) we get

$$[T] = \begin{bmatrix} a_{11}^2 & a_{12}^2 & a_{13}^2 & 2\,a_{12}\,a_{13} & 2\,a_{11}\,a_{13} & 2\,a_{11}\,a_{12} \\ a_{21}^2 & {a_{22}}^2 & a_{23}^2 & 2\,a_{22}\,a_{23} & 2\,a_{21}\,a_{23} & 2\,a_{21}\,a_{22} \\ a_{31}^2 & a_{32}^2 & a_{33}^2 & 2\,a_{32}\,a_{33} & 2\,a_{31}\,a_{33} & 2\,a_{31}\,a_{32} \\ a_{21}\,a_{31} & a_{22}\,a_{32} & a_{23}\,a_{33} & a_{22}\,a_{33} + a_{23}\,a_{32} & a_{21}\,a_{33} + a_{23}\,a_{31} & a_{21}\,a_{32} + a_{22}\,a_{31} \\ a_{11}\,a_{31} & a_{12}\,a_{32} & a_{13}\,a_{33} & a_{12}\,a_{33} + a_{13}\,a_{32} & a_{11}\,a_{33} + a_{13}\,a_{31} & a_{11}\,a_{32} + a_{12}\,a_{31} \\ a_{11}\,a_{21} & a_{12}\,a_{22} & a_{13}\,a_{23} & a_{12}\,a_{23} + a_{13}\,a_{22} & a_{11}\,a_{23} + a_{13}\,a_{21} & a_{11}\,a_{22} + a_{12}\,a_{21} \end{bmatrix} \tag{1.34}$$

A *MATLAB* program that can be used to generate (1.34) is shown next (also available in [5]).

```
% Derivation of the transformation matrix [T]
clear all;
syms T alpha R
syms a a11 a12 a13 a21 a22 a23 a31 a32 a33
a = [a11,a12,a13;
     a21,a22,a23;
     a31,a32,a33];
T(1:6,1:6) = 0;
for i=1:1:3
for j=1:1:3
 if i==j; alpha = j; else alpha = 9-i-j; end
 for p=1:1:3
 for q=1:1:3
  if p==q beta = p; else beta = 9-p-q; end
  T(alpha,beta) = 0;
  if alpha<=3 & beta<= 3; T(alpha,beta)=a(i,p)*a(i,p); end
  if alpha> 3 & beta<= 3; T(alpha,beta)=a(i,p)*a(j,p); end
  if alpha<=3 & beta>3; T(alpha,beta)=a(i,q)*a(i,p)+a(i,p)*a(i,q);end
  if alpha>3 & beta>3; T(alpha,beta)=a(i,p)*a(j,q)+a(i,q)*a(j,p);end
 end
 end
end
end
T
R = eye(6,6); R(4,4)=2; R(5,5)=2; R(6,6)=2; % Reuter matrix
Tbar = R*T*R^(-1)
```

1.10.2 Strain Transformation

The tensor components of strain ε_{ij} transform in the same way as the stress components

$$\varepsilon'_{ij} = a_{ip}a_{jq}\varepsilon_{pq} \tag{1.35}$$

or

$$\varepsilon'_{\alpha} = T_{\alpha\beta}\varepsilon_{\beta} \tag{1.36}$$

with $T_{\alpha\beta}$ given by (1.34). However, the three engineering shear strains $\gamma_{xz}, \gamma_{yz}, \gamma_{xy}$ are normally used instead of tensor shear strains $\varepsilon_{xz}, \varepsilon_{yz}, \varepsilon_{xy}$. The engineering strains (ϵ instead of ε) are defined in (1.5). They can be obtained from the tensor components by the following relationship

$$\epsilon_{\delta} = R_{\delta\gamma}\varepsilon_{\gamma} \tag{1.37}$$

with the Reuter matrix given by

$$[R] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix} \tag{1.38}$$

Then, the coordinate transformation of engineering strain results from (1.36) and (1.37) as

$$\epsilon'_\alpha = \overline{T}_{\alpha\beta}\epsilon_\beta \tag{1.39}$$

with

$$\left[\,\overline{T}\,\right] = [R][T][R]^{-1} \tag{1.40}$$

used only to transform engineering strains. Explicitly we have

$$\left[\,\overline{T}\,\right] =$$

$$\begin{bmatrix} a_{11}^2 & a_{12}^2 & a_{13}^2 & a_{12}\,a_{13} & a_{11}\,a_{13} & a_{11}\,a_{12} \\ a_{21}^2 & a_{22}^2 & a_{23}^2 & a_{22}\,a_{23} & a_{21}\,a_{23} & a_{21}\,a_{22} \\ a_{31}^2 & a_{32}^2 & a_{33}^2 & a_{32}\,a_{33} & a_{31}\,a_{33} & a_{31}\,a_{32} \\ 2\,a_{21}\,a_{31} & 2\,a_{22}\,a_{32} & 2\,a_{23}\,a_{33} & a_{22}\,a_{33} + a_{23}\,a_{32} & a_{21}\,a_{33} + a_{23}\,a_{31} & a_{21}\,a_{32} + a_{22}\,a_{31} \\ 2\,a_{11}\,a_{31} & 2\,a_{12}\,a_{32} & 2\,a_{13}\,a_{33} & a_{12}\,a_{33} + a_{13}\,a_{32} & a_{11}\,a_{33} + a_{13}\,a_{31} & a_{11}\,a_{32} + a_{12}\,a_{31} \\ 2\,a_{11}\,a_{21} & 2\,a_{12}\,a_{22} & 2\,a_{13}\,a_{23} & a_{12}\,a_{23} + a_{13}\,a_{22} & a_{11}\,a_{23} + a_{13}\,a_{21} & a_{11}\,a_{22} + a_{12}\,a_{21} \end{bmatrix} \tag{1.41}$$

1.11 Transformation of Constitutive Equations

The constitutive equations that relate stress $\boldsymbol{\sigma}$ to strain $\boldsymbol{\varepsilon}$ are defined using tensor strains ($\boldsymbol{\varepsilon}$, not $\boldsymbol{\epsilon}$), as

$$\begin{aligned} \boldsymbol{\sigma}' &= \mathbf{C}' : \boldsymbol{\varepsilon}' \\ \sigma'_{ij} &= C'_{ijkl}\varepsilon'_{kl} \end{aligned} \tag{1.42}$$

where both *tensor* and *index* notations have been used.[4]

For simplicity consider an orthotropic material (Section 1.12.3). Then, it is possible to write σ'_{11}, and σ'_{12} as

$$\begin{aligned} \sigma'_{11} &= C'_{1111}\varepsilon'_{11} + C'_{1122}\varepsilon'_{22} + C'_{1133}\varepsilon'_{33} \\ \sigma'_{12} &= C'_{1212}\varepsilon'_{12} + C'_{1221}\varepsilon'_{21} = 2C'_{1212}\varepsilon'_{12} \end{aligned} \tag{1.43}$$

[4] A double contraction involves contraction of two indices, in this case k and l, and it is denoted by : in tensor notation. Also note the use of boldface to indicate tensors in tensor notation.

Rewriting (1.43) in contracted notation, it is clear that in contracted notation all the shear strains appear twice, as follows

$$\begin{aligned}\sigma'_1 &= C'_{11}\varepsilon'_1 + C'_{12}\varepsilon'_2 + C'_{13}\varepsilon'_3 \\ \sigma'_6 &= 2C'_{66}\varepsilon'_6\end{aligned} \tag{1.44}$$

The factor 2 in front of the tensor shear strains is caused by two facts, the minor symmetry of the tensors C and ε (see (1.5,1.55-1.56)) and the contraction of the last two indices of C_{ijkl} with the strain ε_{kl} in (1.43). Therefore, *any double contraction of tensors with minor symmetry needs to be corrected by a Reuter matrix (1.38) when written in the contracted notation.* Next, (1.42) can be written as

$$\sigma'_\alpha = C'_{\alpha\beta} R_{\beta\delta} \varepsilon'_\delta \tag{1.45}$$

Note that the Reuter matrix in (1.45) can be combined with the tensor strains using (1.37), to write

$$\sigma'_\alpha = C'_{\alpha\beta} \epsilon'_\beta \tag{1.46}$$

in terms of engineering strains. To obtain the stiffness matrix $[C]$ in the laminate coordinate system, introduce (1.29) and (1.39) into (1.46) so that

$$T_{\alpha\delta}\sigma_\delta = C'_{\alpha\beta}\overline{T}_{\beta\gamma}\epsilon_\gamma \tag{1.47}$$

It can be shown that

$$[T]^{-1} = [\overline{T}]^T \tag{1.48}$$

Therefore

$$\{\sigma\} = [C]\{\epsilon\} \tag{1.49}$$

with

$$[C] = [\overline{T}]^T [C'][\overline{T}] \tag{1.50}$$

and

$$[C'] = [\overline{T}]^{-T}[C][\overline{T}]^{-1} = [T][C][T]^T \tag{1.51}$$

The compliance matrix is the inverse of the stiffness matrix, not the inverse of the fourth order tensor C_{ijkl}. Therefore,

$$[S'] = [C']^{-1} \tag{1.52}$$

Taking into account (1.48) and (1.50), the compliance matrix transforms as

$$[S] = [T]^T [S'][T] \tag{1.53}$$

$$[S'] = [T]^{-T}[S][T]^{-1} = [\overline{T}][S][\overline{T}]^T \tag{1.54}$$

1.12 3D Constitutive Equations

Hooke's law in three dimensions (3D) takes the form of (1.42). The 3D stiffness tensor C_{ijkl} is a fourth-order tensor with 81 components. For anisotropic materials only 21 components are independent. That is, the remaining 60 components can be written in terms of the other 21. The one-dimensional case (1D), studied in mechanics of materials, is recovered when all the stress components are zero except σ_{11}. Only for the 1D case, $\sigma_{11} = \sigma, \varepsilon_{11} = \epsilon, C_{1111} = E$, and $\sigma = E\epsilon$. All the derivations in this section are carried out in lamina coordinates *but for simplicity the prime symbol (′) is omitted, in this section only.*

In (1.42), exchanging the dummy indexes i by j, and k by l we have

$$\sigma_{ji} = C_{jilk}\varepsilon_{lk} \tag{1.55}$$

Since the stress and strain tensors are symmetric, i.e, $\sigma_{ij} = \sigma_{ji}$ and $\varepsilon_{kl} = \varepsilon_{kl}$, it follows that

$$C_{ijkl} = C_{jikl} = C_{ijlk} = C_{jilk} \tag{1.56}$$

which effectively reduces the number of independent components from 81 to 36. For example, $C_{1213} = C_{2131}$ and so on. Then, the 36 independent components can be written as a 6×6 matrix.

Furthermore, an elastic material does not dissipate energy. All elastic energy stored during loading is recovered during unloading. Therefore, the elastic energy at any point on the stress-strain curve is independent on the path that was followed to arrive at that point. A path independent function is called a potential function. In this case, the potential is the strain energy density $\widetilde{u}(\varepsilon_{ij})$. Expanding the strain energy density in a Taylor power series

$$\widetilde{u} = \widetilde{u}_0 + \left.\frac{\partial \widetilde{u}}{\partial \varepsilon_{ij}}\right|_0 \varepsilon_{ij} + \frac{1}{2} \left.\frac{\partial^2 \widetilde{u}}{\partial \varepsilon_{ij} \partial \varepsilon_{kl}}\right|_0 \varepsilon_{ij}\, \varepsilon_{kl} + ... \tag{1.57}$$

Now take a derivative with respect to ε_{ij}

$$\frac{\partial \widetilde{u}}{\partial \varepsilon_{ij}} = 0 + \beta_{ij} + \frac{1}{2} \left(\alpha_{ijkl}\, \varepsilon_{kl} + \alpha_{klij}\, \varepsilon_{ij}\right) \tag{1.58}$$

where β_{ij} and α_{ijkl} are constants. From here, one can write

$$\sigma_{ij} - \sigma^0_{ij} = C_{ijkl}\, \varepsilon_{kl} \tag{1.59}$$

where $\sigma^0_{ij} = \beta_{ij}$ is the residual stress and $\alpha_{ijkl} = 1/2(C_{ijkl} + C_{klik}) = C_{ijkl}$ is the symmetric stiffness tensor (see (1.56)). Equation (1.59) is a generalization of (1.55) including residual stresses.

Using contracted notation, the generalized Hooke's law becomes

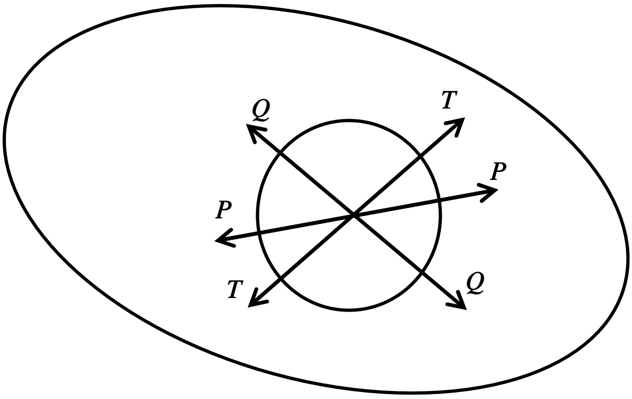


Fig. 1.9: Anisotropic material.

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{Bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ C_{12} & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} \\ C_{13} & C_{23} & C_{33} & C_{34} & C_{35} & C_{36} \\ C_{14} & C_{24} & C_{34} & C_{44} & C_{45} & C_{46} \\ C_{15} & C_{25} & C_{35} & C_{45} & C_{55} & C_{56} \\ C_{16} & C_{26} & C_{36} & C_{46} & C_{56} & C_{66} \end{bmatrix} \begin{Bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \gamma_4 \\ \gamma_5 \\ \gamma_6 \end{Bmatrix} \tag{1.60}$$

Once again, the 1D case is covered when $\sigma_\alpha = 0$ if $\alpha \neq 1$. Then, $\sigma_1 = \sigma, \epsilon_1 = \epsilon, C_{11} = E$.

1.12.1 Anisotropic Material

Equation (1.60) represents a fully anisotropic material. Such a material has properties that change with the orientation. For example, the material body depicted in Figure 1.9 deforms differently in the directions P, T, and Q, even if the forces applied along the directions P, T, and Q are equal. The number of constants required to describe anisotropic materials is 21.

The inverse of the stiffness matrix is the compliance matrix $[S] = [C]^{-1}$. The constitutive equation (3D Hooke's law) is written in terms of compliances as follows

$$\begin{Bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \gamma_4 \\ \gamma_5 \\ \gamma_6 \end{Bmatrix} = \begin{bmatrix} S_{11} & S_{12} & S_{13} & S_{14} & S_{15} & S_{16} \\ S_{12} & S_{22} & S_{23} & S_{24} & S_{25} & S_{26} \\ S_{13} & S_{23} & S_{33} & S_{34} & S_{35} & S_{36} \\ S_{14} & S_{24} & S_{34} & S_{44} & S_{45} & S_{46} \\ S_{15} & S_{25} & S_{35} & S_{45} & S_{55} & S_{56} \\ S_{16} & S_{26} & S_{36} & S_{46} & S_{56} & S_{66} \end{bmatrix} \begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{Bmatrix} \tag{1.61}$$

The [S] matrix is also symmetric and it has 21 independent constants. For the 1D case, $\sigma = 0$ if $p \neq 1$. Then, $\sigma_1 = \sigma, \epsilon_1 = \epsilon, S_{11} = 1/E$.

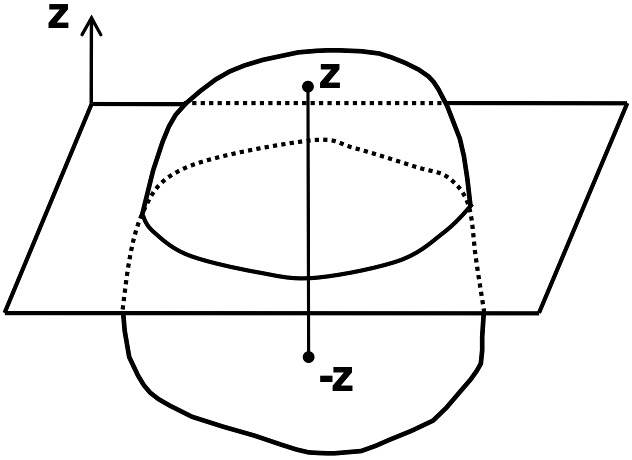


Fig. 1.10: Monoclinic material.

1.12.2 Monoclinic Material

If a material has one plane of symmetry (Figure 1.10) it is called monoclinic and 13 constants are required to describe it. One plane of symmetry means that the properties are the same at symmetric points (z and -z as in Figure 1.10).

When the material is symmetric about the 1-2 plane, the material properties are identical upon reflection with respect to the 1-2 plane. For such reflection the a-matrix (1.21) is

$$\begin{array}{c c} & \begin{array}{ccc} x_1 & x_2 & x_3 \end{array} \\ \begin{array}{c} e_1'' \\ e_2'' \\ e_3'' \end{array} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \end{array} \tag{1.62}$$

where $''$ has been used to avoid confusion with the lamina coordinate system that is denoted without $()'$ in this Section but with $()'$ elsewhere in this book. From (1.40) we get

$$[\overline{T}] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.63}$$

The effect of $[\overline{T}]$ is to multiply rows and columns 4 and 5 in $[C]$ by -1. The diagonal terms C_{44} and C_{55} remain positive because they are multiplied twice. Therefore, $C''_{i4} = -C_{i4}$ *with* $i \neq 4,5$, $C''_{i5} = -C_{i5}$ *with* $i \neq 4,5$, with everything else unchanged. Since the material properties in a monoclinic material cannot change by a reflection, it must be $C_{4i} = C_{i4} = 0$ *with* $i \neq 4,5$, $C_{5i} = C_{i5} = 0$ *with*

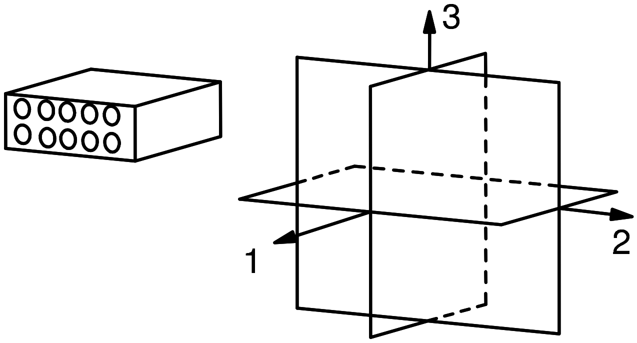


Fig. 1.11: Orthotropic material.

$i \neq 4, 5$. That is, 3D Hooke's law reduces to

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{Bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & 0 & 0 & C_{16} \\ C_{12} & C_{22} & C_{23} & 0 & 0 & C_{26} \\ C_{13} & C_{23} & C_{33} & 0 & 0 & C_{36} \\ 0 & 0 & 0 & C_{44} & C_{45} & 0 \\ 0 & 0 & 0 & C_{45} & C_{55} & 0 \\ C_{16} & C_{26} & C_{36} & 0 & 0 & C_{66} \end{bmatrix} \begin{Bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \gamma_4 \\ \gamma_5 \\ \gamma_6 \end{Bmatrix} \tag{1.64}$$

and in terms of the compliances to

$$\begin{Bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \gamma_4 \\ \gamma_5 \\ \gamma_6 \end{Bmatrix} = \begin{bmatrix} S_{11} & S_{12} & S_{13} & 0 & 0 & S_{16} \\ S_{12} & S_{22} & S_{23} & 0 & 0 & S_{26} \\ S_{13} & S_{23} & S_{33} & 0 & 0 & S_{36} \\ 0 & 0 & 0 & S_{44} & S_{45} & 0 \\ 0 & 0 & 0 & S_{45} & S_{55} & 0 \\ S_{16} & S_{26} & S_{36} & 0 & 0 & S_{66} \end{bmatrix} \begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{Bmatrix} \tag{1.65}$$

1.12.3 Orthotropic Material

An orthotropic material has three planes of symmetry that coincide with the coordinate planes. It can be shown that if two orthogonal planes of symmetry exist, there is always a third orthogonal plane of symmetry. Nine constants are required to describe this type of material.

The symmetry planes can be Cartesian, as depicted in Figure 1.11, or they may correspond to any other coordinate representation (cylindrical, spherical, etc.). For example, the trunk of a tree has cylindrical orthotropy because of the growth rings. However, most practical materials exhibit Cartesian orthotropy. A unidirectional fiber reinforced composite may be considered to be orthotropic. One plane of symmetry is perpendicular to the fiber direction, and the other two are parallel to the fiber direction and orthogonal among themselves.

In addition to the reflection about the 1-2 plane discussed in Section 1.12.2, a second reflection about the 1-3 plane should not affect the properties of the orthotropic materials. In this case the a-matrix is

$$[a] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1.66}$$

The $\overline{T}$-matrix from (1.40) is

$$[\overline{T}] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \tag{1.67}$$

This will make $C_{i6} = -C_{i6}$, $i \neq 4, 6$ and $C_{i4} = -C_{i4}$, $i \neq 4, 6$. Since the material has symmetry about the 1-3 plane, this means that $C_{i6} = C_{6i} = 0$, $i \neq 6$. In this case, 3D Hooke's law reduces to

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{Bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & 0 & 0 & 0 \\ C_{12} & C_{22} & C_{23} & 0 & 0 & 0 \\ C_{13} & C_{23} & C_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{66} \end{bmatrix} \begin{Bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \gamma_4 \\ \gamma_5 \\ \gamma_6 \end{Bmatrix} \tag{1.68}$$

and in terms of the compliances to

$$\begin{Bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \gamma_4 \\ \gamma_5 \\ \gamma_6 \end{Bmatrix} = \begin{bmatrix} S_{11} & S_{12} & S_{13} & 0 & 0 & 0 \\ S_{12} & S_{22} & S_{23} & 0 & 0 & 0 \\ S_{13} & S_{23} & S_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & S_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & S_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & S_{66} \end{bmatrix} \begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{Bmatrix} \tag{1.69}$$

Note that if the material has two planes of symmetry, it automatically has three because applying the procedure once more for a third plane (the 2-3 plane) will not change (1.68-1.69).

1.12.4 Transversely Isotropic Material

A transversely isotropic material has one axis of symmetry. For example, the fiber direction of a unidirectional fiber reinforced composite can be considered an axis of symmetry if the fibers are randomly distributed in the cross section (Figure 1.12). In this case, any plane containing the fiber direction is a plane of symmetry. A transversely isotropic material is described by five constants. When the axis of symmetry is the fiber direction (1-direction), 3D Hooke's law reduces to

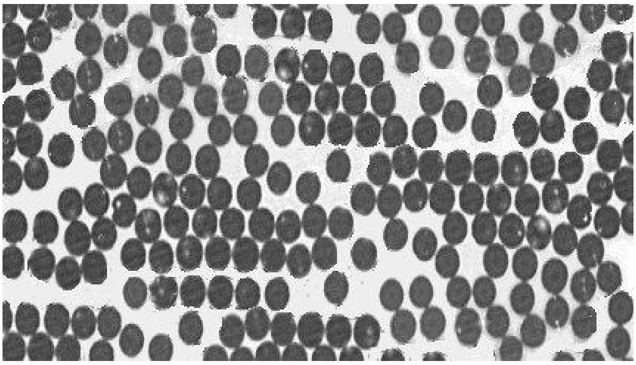


Fig. 1.12: Randomly distributed E-glass fibers with 200X magnification.

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{Bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{22} & C_{23} & 0 & 0 & 0 \\ C_{12} & C_{23} & C_{22} & 0 & 0 & 0 \\ 0 & 0 & 0 & (C_{22}-C_{23})/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{66} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{66} \end{bmatrix} \begin{Bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \gamma_4 \\ \gamma_5 \\ \gamma_6 \end{Bmatrix} \tag{1.70}$$

and in terms of the compliances to

$$\begin{Bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \gamma_4 \\ \gamma_5 \\ \gamma_6 \end{Bmatrix} = \begin{bmatrix} S_{11} & S_{12} & S_{12} & 0 & 0 & 0 \\ S_{12} & S_{22} & S_{23} & 0 & 0 & 0 \\ S_{12} & S_{23} & S_{22} & 0 & 0 & 0 \\ 0 & 0 & 0 & 2(S_{22}-S_{23}) & 0 & 0 \\ 0 & 0 & 0 & 0 & S_{66} & 0 \\ 0 & 0 & 0 & 0 & 0 & S_{66} \end{bmatrix} \begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{Bmatrix} \tag{1.71}$$

Note the equations would be different if the axis of symmetry is not the 1-direction. In terms of engineering properties (Section 1.13), and taking into account that the directions 2 and 3 are indistinguishable, the following relations apply for a transversely isotropic material:

$$\begin{aligned} E_2 &= E_3 \\ \nu_{12} &= \nu_{13} \\ G_{12} &= G_{13} \end{aligned} \tag{1.72}$$

In addition, any two perpendicular directions on the plane 2-3 can be taken as axes. In other words, the plane 2-3 is isotropic. Therefore, the following holds in the 2-3 plane

$$G_{23} = \frac{E_2}{2(1+\nu_{23})} \tag{1.73}$$

just as it holds for isotropic materials (see Problem 1.14).

1.12.5 Isotropic Material

The most common materials of industrial use are isotropic, like aluminum, steel, etc. Isotropic materials have an infinite number of planes of symmetry, meaning that the properties are independent of the orientation. Only two constants are needed to represent the elastic properties. These two properties can be the Young's modulus E and the Poisson's ratio ν, but several other pairs of constants are used whenever it is convenient. However, any pair of properties has to be related to any other pair. For example, you could describe isotropic materials by E and G, but the shear modulus of isotropic materials is related to E and ν by

$$G = \frac{E}{2(1+\nu)} \tag{1.74}$$

Also, the Lamé constants are sometimes used for convenience, in this case the two constants are

$$\begin{aligned} \lambda &= \frac{E}{(1+\nu)(1-2\nu)} \\ \mu &= G \end{aligned} \tag{1.75}$$

To form yet another pair, any of the above properties could be substituted by the bulk modulus k, as follows

$$k = \frac{E}{3(1-2\nu)} \tag{1.76}$$

which relates the hydrostatic pressure p to the volumetric strain as

$$p = k(\epsilon_{1+}\epsilon_{2+}\epsilon_3) \tag{1.77}$$

For isotropic materials, the 3D Hooke's law is written in terms of only two constants C_{11} and C_{12} as

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{Bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{11} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{12} & C_{11} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{(C_{11}-C_{12})}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{(C_{11}-C_{12})}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{(C_{11}-C_{12})}{2} \end{bmatrix} \begin{Bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \gamma_4 \\ \gamma_5 \\ \gamma_6 \end{Bmatrix} \tag{1.78}$$

In terms of compliances, once again, two constants are used, S_{11} and S_{12}, as follows

$$
\begin{aligned}
\left\{\begin{array}{c}\epsilon_1\\ \epsilon_2\\ \epsilon_3\\ \gamma_4\\ \gamma_5\\ \gamma_6\end{array}\right\} &= \left[\begin{array}{cccccc}S_{11} & S_{12} & S_{12} & 0 & 0 & 0\\ S_{12} & S_{11} & S_{12} & 0 & 0 & 0\\ S_{12} & S_{12} & S_{11} & 0 & 0 & 0\\ 0 & 0 & 0 & 2s & 0 & 0\\ 0 & 0 & 0 & 0 & 2s & 0\\ 0 & 0 & 0 & 0 & 0 & 2s\end{array}\right] \left\{\begin{array}{c}\sigma_1\\ \sigma_2\\ \sigma_3\\ \sigma_4\\ \sigma_5\\ \sigma_6\end{array}\right\} \\
s &= S_{11} - S_{12}
\end{aligned}
\tag{1.79}
$$

Not only are the various constants related in pairs, but also certain restrictions apply on the values that these constants may have for real materials. Since the Young and shear moduli must always be positive, the Poisson's ratio must be $\nu > -1$. Furthermore, since the bulk modulus must be positive, we have $\nu < \frac{1}{2}$. Finally, the Poisson's ratio of isotropic materials is constrained by $-1 < \nu < \frac{1}{2}$.

1.13 Engineering Constants

Please note from here forward $()'$ denotes the lamina coordinate system. Our next task is to write the components of the stiffness and compliance matrices in terms of engineering constants for orthotropic materials. For this purpose it is easier to work with the compliance matrix, which is defined as the inverse of the stiffness matrix. In *lamina coordinates* $[S'] = [C']^{-1}$. The compliance matrix is used to write the relationship between strains and stresses in (1.69) for an orthotropic material. Let's rewrite the first of (1.69), which corresponds to the strain in the 1-direction (fiber direction)

$$\epsilon_1' = S_{11}'\sigma_1' + S_{12}'\sigma_2' + S_{13}'\sigma_3' \tag{1.80}$$

and let's perform a thought experiment. Note that $[S']$ is used to emphasize the fact that we are working in the lamina coordinate system. First, apply a tensile stress along the 1-direction (fiber direction) as in Figure 1.13, with all the other stresses equal to zero, and compute the strain produced in the 1-direction, which is

$$\epsilon_1' = \frac{{\sigma_1}'}{E_1} \tag{1.81}$$

Then, apply a stress in the 2-direction only, and compute the strain in the 1-direction using the appropriate Poisson's ratio [1]

$$\epsilon_1' = -\nu_{21}\frac{\sigma_2'}{E_2} \tag{1.82}$$

Now, apply a stress in the 3-direction only, and compute the strain in the 1-direction using the appropriate Poisson's ratio,

$$\epsilon_1' = -\nu_{31}\frac{\sigma_3'}{E_3} \tag{1.83}$$

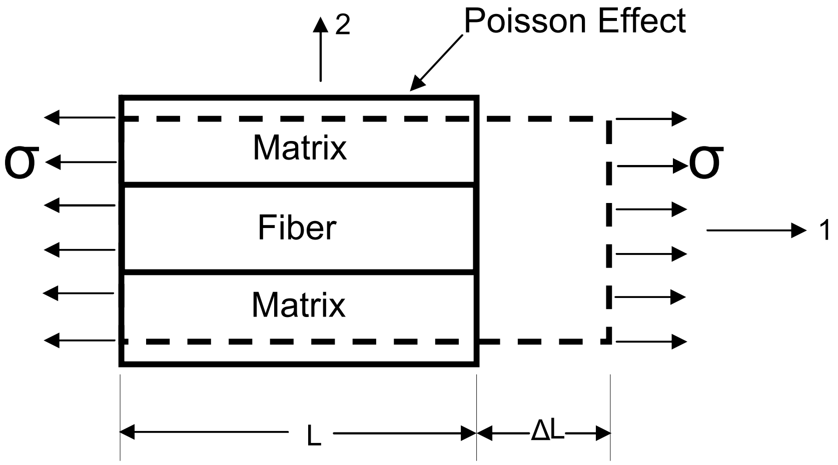


Fig. 1.13: Longitudinal loading.

The total strain ϵ_1' is the sum of equations (1.81), (1.82), and (1.83)

$$\epsilon_1' = \frac{1}{E_1}\sigma_1' - \frac{\nu_{21}}{E_2}\sigma_2' - \frac{\nu_{31}}{E_3}\sigma_3' \tag{1.84}$$

Comparing (1.84) with (1.80) we conclude that

$$S_{11}' = \frac{1}{E_1}; S_{12}' = -\frac{\nu_{21}}{E_2}; S_{13}' = -\frac{\nu_{31}}{E_3} \tag{1.85}$$

Repeat the same procedure for the equations corresponding to ϵ_2' and ϵ_3' to obtain the coefficients in the second and third rows of the compliance matrix (1.69).

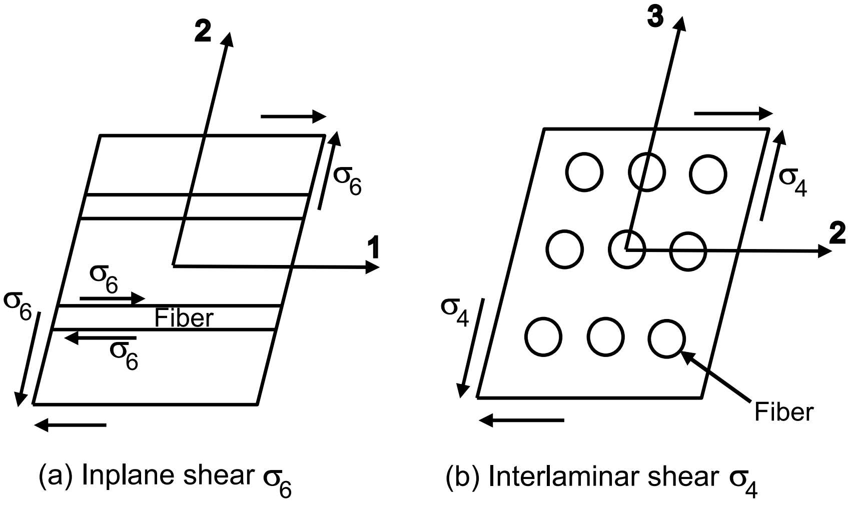
For the shear terms use the 4th, 5th, and 6th rows of the compliance matrix (1.69). For example, from Figure 1.14 we write

$$\sigma_6' = \epsilon_6' G_{12} = 2\varepsilon_6' G_{12} \tag{1.86}$$

which compared to the 6th row of (1.69) leads to $S_{66} = 1/G_{12}$.

For an orthotropic material, the compliance matrix $[S']$ is defined in the lamina coordinate system as

$$[S'] = \begin{bmatrix} \frac{1}{E_1} & \frac{-\nu_{21}}{E_2} & \frac{-\nu_{31}}{E_3} & 0 & 0 & 0 \\ \frac{-\nu_{12}}{E_1} & \frac{1}{E_2} & \frac{-\nu_{32}}{E_3} & 0 & 0 & 0 \\ \frac{-\nu_{13}}{E_1} & \frac{-\nu_{23}}{E_2} & \frac{1}{E_3} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{23}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{13}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{12}} \end{bmatrix} \tag{1.87}$$



(a) Inplane shear σ_6 (b) Interlaminar shear σ_4

Fig. 1.14: Shear loading.

where E_i, G_{ij}, and ν_{ij} are the elastic moduli, shear moduli, and Poisson's ratios, respectively. Furthermore, the subscripts indicate lamina coordinates, i.e.,

$$\nu_{ij} = \nu_{x'_i x'_j} \text{ and } E_{ii} = E_{x'_i} \tag{1.88}$$

Since $[S']$ is symmetric, the following must be satisfied

$$\frac{\nu_{ij}}{E_{ii}} = \frac{\nu_{ji}}{E_{jj}} \text{ , } i,j = 1\dots 3 \tag{1.89}$$

Furthermore, Poisson's ratios are defined so that the lateral strain is given by

$$\nu_j = -\nu_{ij}\epsilon_i \tag{1.90}$$

In *ANSYS*, the Poisson's ratios are defined differently than in this textbook. In fact, $\nu_{xy}, \nu_{xz}, \nu_{yz}$ are denoted PRXY, PRXZ, and PRYZ, while $\nu_{yx}, \nu_{zx}, \nu_{zy}$ are denoted by NUXY, NUXZ, and NUYZ. On the contrary, *Abaqus* uses the standard notation also used in this textbook. That is, the symbols NU12, NU13, NU23 follow the convention described by (1.90).

After computing S_{ij}, the components of stress are obtained by using (1.46) or (1.49). This formulation predicts realistic behavior for finite displacement and rotations as long as the strains are small. This formulation is expensive to use since it needs 18 state variables: 12 components of the strain displacement matrix computed in the initial configuration ($u_{i,j}$ and $u_{r,i}u_{r,j}$) plus 6 direction cosines $[a]$ to account for finite rotations.

However, in (1.87) only nine constants are independent because the matrix $[S']$ must be symmetric (see 1.93), so

$$[S'] = \begin{bmatrix} \frac{1}{E_1} & -\frac{\nu_{12}}{E_1} & -\frac{\nu_{13}}{E_1} & 0 & 0 & 0 \\ -\frac{\nu_{12}}{E_1} & \frac{1}{E_2} & -\frac{\nu_{23}}{E_2} & 0 & 0 & 0 \\ -\frac{\nu_{13}}{E_1} & -\frac{\nu_{23}}{E_2} & \frac{1}{E_3} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{23}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{13}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{12}} \end{bmatrix} \tag{1.91}$$

The stiffness matrix can be computed also in terms of engineering constants by inverting the above equation so that $[C'] = [S']^{-1}$, with components given in terms of engineering constants as

$$\begin{aligned} C'_{11} &= \frac{1-\nu_{23}\nu_{32}}{E_2E_3\Delta} \\ C'_{12} &= \frac{\nu_{21}+\nu_{31}\nu_{23}}{E_2E_3\Delta} = \frac{\nu_{12}+\nu_{32}\nu_{13}}{E_1E_3\Delta} \\ C'_{13} &= \frac{\nu_{31}+\nu_{21}\nu_{32}}{E_2E_3\Delta} = \frac{\nu_{13}+\nu_{12}\nu_{23}}{E_1E_2\Delta} \\ C'_{22} &= \frac{1-\nu_{13}\nu_{31}}{E_1E_3\Delta} \\ C'_{23} &= \frac{\nu_{32}+\nu_{12}\nu_{31}}{E_1E_3\Delta} = \frac{\nu_{23}+\nu_{21}\nu_{13}}{E_1E_2\Delta} \\ C'_{33} &= \frac{1-\nu_{12}\nu_{21}}{E_1E_2\Delta} \\ C'_{44} &= G_{23} \\ C'_{55} &= G_{13} \\ C'_{66} &= G_{12} \qquad (1.92) \\ \Delta &= \frac{1-\nu_{12}\nu_{21}-\nu_{23}\nu_{32}-\nu_{31}\nu_{13}-2\nu_{21}\nu_{32}\nu_{13}}{E_1E_2E_3} \end{aligned}$$

So far both $[S']$ and $[C']$ are 6×6 matrices with 9 independent constants for the case of orthotropic materials. If the material is transversely isotropic $G_{13} = G_{12}, \nu_{13} = \nu_{12}, E_3 = E_2$.

1.13.1 Restrictions on Engineering Constants

It is important to note that because of the symmetry of the compliance matrix (1.91), the following restrictions on engineering constants apply

$$\frac{\nu_{ij}}{E_i} = \frac{\nu_{ji}}{E_j}; \quad i,j = 1\ldots 3; \quad i \neq j \tag{1.93}$$

Further restrictions on the values of the elastic constants can be derived from the fact that all diagonal terms in both the compliance and stiffness matrices must be positive. Since all the engineering elastic constants must be positive $(E_1, E_2, E_3, G_{12}, G_{23}, G_{31} > 0)$, all the diagonal terms of the stiffness matrix (1.92) will be positive if the following two conditions are met. The first condition is that $(1 - \nu_{ij}\nu_{ji}) > 0$ for $i, j = 1..3$ and $i \neq j$, which leads to the following restriction on the values of the engineering constants

$$0 < \nu_{ij} < \sqrt{\frac{E_i}{E_j}};\ i, j = 1 \ldots 3;\ i \neq j \tag{1.94}$$

The second condition is that

$$\Delta = 1 - \nu_{12}\nu_{21} - \nu_{23}\nu_{32} - \nu_{31}\nu_{13} - 2\nu_{21}\nu_{32}\nu_{13} > 0 \tag{1.95}$$

These restrictions can be used to check experimental data. For example, consider an experimental program in which if E_1 and ν_{12} are measured in a longitudinal test (fibers in the direction of loading) by using two strain gages, one longitudinal and one transverse, and E_2 and ν_{21} are measured in the transverse tensile tests (fibers perpendicular to loading). For the test procedure to be valid, all the four data values, E_1,E_2, ν_{12}, and ν_{21}, must conform to (1.93-1.95) within the margin allowed by experimental errors.

Example 1.4 *Sonti et al. [6] performed a series of tests on pultruded glass-fiber reinforced composites. From tensile tests along the longitudinal axis, the average of eight tests gives* $E_1 = 19.981\ GPa$ *and* $\nu_{12} = 0.274$. *The average of eight tests in the transverse direction gives* $E_2 = 11.389\ GPa$ *and* $\nu_{21} = 0.192$. *Do these data fall within the constraints on elastic constants?*

Solution to Example 1.4 *First compute both sides of (1.93) for* $i, j = 1, 2$ *as*

$$\frac{E_1}{\nu_{12}} = \frac{19.981}{0.274} = 72.9\ GPa$$
$$\frac{E_2}{\nu_{21}} = \frac{11.389}{0.192} = 59.3\ GPa$$

The transverse result is 23% lower than expected. Either E_2 *measured is too low or* ν_{21} *measured is 23% higher than what it should be. In any case a 23% difference deserves some scrutiny.*

Next check (1.94)

$$abs(\nu_{12}) < \sqrt{\frac{E_1}{E_2}}$$
$$0.274 < 1.32$$
$$abs(\nu_{21}) < \sqrt{\frac{E_2}{E_1}}$$
$$0.192 < 0.75$$

Finally, there are insufficient data to evaluate the last of the restrictions on elastic constants from (1.95).

1.14 From 3D to Plane Stress Equations

Setting $\sigma_3 = 0$ in the compliance equations (1.69) of an orthotropic material implies that the third row and column of the compliance matrix are not used

$$\left\{\begin{array}{c}\epsilon'_1\\ \epsilon'_2\\ \epsilon'_3\\ \gamma'_4\\ \gamma'_5\\ \gamma'_6\end{array}\right\} = \left[\begin{array}{cccccc} S'_{11} & S'_{12} & S'_{13} & 0 & 0 & 0\\ S'_{12} & S'_{22} & S'_{23} & 0 & 0 & 0\\ S'_{13} & S'_{23} & S'_{33} & 0 & 0 & 0\\ 0 & 0 & 0 & S'_{44} & 0 & 0\\ 0 & 0 & 0 & 0 & S'_{55} & 0\\ 0 & 0 & 0 & 0 & 0 & S'_{66}\end{array}\right] \left\{\begin{array}{c}\sigma'_1\\ \sigma'_2\\ \sigma'_3 = 0\\ \sigma'_4\\ \sigma'_5\\ \sigma'_6\end{array}\right\} \tag{1.96}$$

So, the first two equations plus the last one can be written separately of the remaining, in terms of a 3×3 reduced compliance matrix $[S]$ and using $\gamma = 2\epsilon$, we have

$$\left\{\begin{array}{c}\epsilon'_1\\ \epsilon'_2\\ \gamma'_6\end{array}\right\} = \left[\begin{array}{ccc} S'_{11} & S'_{12} & 0\\ S'_{12} & S'_{22} & 0\\ 0 & 0 & S'_{66}\end{array}\right] \left\{\begin{array}{c}\sigma'_1\\ \sigma'_2\\ \sigma'_6\end{array}\right\} \tag{1.97}$$

The third equation is seldom used

$$\epsilon'_3 = S'_{13}\sigma'_1 + S'_{23}\sigma'_2 \tag{1.98}$$

and the remaining two equations can be written separately as

$$\left\{\begin{array}{c}\gamma'_4\\ \gamma'_5\end{array}\right\} = \left[\begin{array}{cc} S'_{44} & 0\\ 0 & S'_{55}\end{array}\right] \left\{\begin{array}{c}\sigma'_4\\ \sigma'_5\end{array}\right\} \tag{1.99}$$

To compute stress components from strains, (1.97) can be inverted to get $\{\sigma\} = [Q]\{\epsilon\}$ or

$$\left\{\begin{array}{c}\sigma'_1\\ \sigma'_2\\ \sigma'_6\end{array}\right\} = \left[\begin{array}{ccc} Q'_{11} & Q'_{12} & 0\\ Q'_{12} & Q'_{22} & 0\\ 0 & 0 & Q'_{66}\end{array}\right] \left\{\begin{array}{c}\epsilon'_1\\ \epsilon'_2\\ \gamma'_6\end{array}\right\} \tag{1.100}$$

where the matrix $[Q'] = [S'_{3\times3}]^{-1}$ is the reduced stiffness matrix for plane stress. Note that while the components of the reduced compliance matrix $[S'_{3\times3}]$ are numerically identical to the corresponding entries in the 6×6 compliance matrix, the components of the reduced stiffness matrix $[Q']$ are not numerically equal to the corresponding entries on the 6×6 stiffness matrix $[C']$, thus the change in name. This is because the inverse of a 3×3 matrix produces different values than the inverse of a 6×6 matrix. The set of equations is completed by writing

$$\left\{\begin{array}{c}\sigma'_4\\ \sigma'_5\end{array}\right\} = \left[\begin{array}{cc} C'_{44} & 0\\ 0 & C'_{55}\end{array}\right] \left\{\begin{array}{c}\gamma'_4\\ \gamma'_6\end{array}\right\} \tag{1.101}$$

where the coefficient C'_{44} and C'_{55} are numerically equal to the corresponding entries in the 6×6 stiffness matrix because the 2×2 matrix in (1.101) is diagonal.

Example 1.5 *Show that the change in the thickness $t\epsilon_3$ of a plate is negligible when compared to the in-plane elongations $a\epsilon_1$ and $b\epsilon_2$. Use the data from a composite plate with thickness $t = 0.635\ mm$, and dimensions $a = 279\ mm$ and $b = 203\ mm$. Take $E_1 = 19.981\ GPa$, $E_2 = 11.389\ GPa$, $\nu_{12} = 0.274$.*

Solution to Example 1.5 *Assuming that the 0.635 mm thick glass-reinforced Polyester plate is transversely isotropic, take $E_3 = E_2 = 11.389\ GPa$, $\nu_{13} = \nu_{12} = 0.274$, $G_{31} = G_{12}$. Sonti et al. [6] report the average of eight torsion tests as $G_{12} = 3.789\ GPa$. Lacking experimental data, assume $\nu_{23} \approx \nu_m = 0.3$, $G_{23} \approx G_m = 0.385\ GPa$, with the properties of the Polyester matrix taken from [1]. The remaining properties in (1.91) can be obtained, using (1.93), as*

$$
\begin{aligned}
\nu_{21} &= \nu_{12}\frac{E_2}{E_1} = 0.274\left(\frac{11.389}{19.981}\right) = 0.156\\
\nu_{31} &= \nu_{13}\frac{E_3}{E_1} = 0.274\left(\frac{11.389}{19.981}\right) = 0.156\\
\nu_{32} &= \nu_{23}\frac{E_3}{E_2} = 0.3\left(\frac{11.389}{11.389}\right) = 0.3
\end{aligned}
$$

Because of transverse isotropy $G_{13} = G_{12} = 3.789\ GPa$. Now, assume a state of stress $\sigma'_1 = \sigma'_2 = 0.1\ GPa, \sigma'_4 = \sigma'_5 = \sigma'_6 = 0$, and $\sigma'_3 = 0$ because of the assumption of plane stress. Using (1.97) we get

$$
\begin{aligned}
\epsilon'_1 &= S'_{11}\sigma'_1 + S'_{12}\sigma'_2 = \frac{0.1}{19.981} - \frac{0.1(0.156)}{11.389} = 3.635\ 10^{-3}\\
\epsilon'_2 &= S'_{12}\sigma'_1 + S'_{22}\sigma'_2 = -\frac{0.1(0.156)}{11.389} + \frac{0.1}{11.389} = 7.411\ 10^{-3}\\
\epsilon'_3 &= S'_{13}\sigma'_1 + S'_{23}\sigma'_2 = -\frac{0.274(0.1)}{19.981} - \frac{0.3(0.1)}{11.389} = -4.005\ 10^{-3}
\end{aligned}
$$

Finally

$$
\begin{aligned}
t\epsilon'_3 &= -0.635(4.005\ 10^{-3}) = -2.543\ 10^{-3}\ mm\\
a\epsilon'_1 &= 279(3.635\ 10^{-3}) = 1.014\ mm\\
b\epsilon'_2 &= 203(7.411\ 10^{-3}) = 1.504\ mm
\end{aligned}
$$

Since the elongation in the transverse direction is so small, it is neglected in the derivation of the plate equations in [1].

1.15 Apparent Laminate Properties

The stiffness matrix $[C]$ of a balanced, symmetric laminate with N layers is built by adding the lamina matrices in laminate coordinate system multiplied by the thickness ratio t_k/t of each layer, where t is the laminate thickness and t_k denotes the thickness of the k-th layer

$$
[C] = \sum_{k=1}^{N} \frac{t_k}{t}[C_k] \tag{1.102}
$$

Note that compliances cannot be added nor averaged. The laminate compliance is obtained inverting the 6×6 stiffness matrix, as

$$[S] = [C]^{-1} \tag{1.103}$$

A laminate is called balanced if the total thickness of layers oriented with respect to the laminate direction at $+\theta$ and $-\theta$ are the same. Such laminate has orthotropic stiffness $[C]$ and compliance $[S]$. In terms of the apparent engineering properties of the laminate, the compliance is

$$[S] = \begin{bmatrix} \dfrac{1}{E_x} & -\dfrac{\nu_{yx}}{E_y} & -\dfrac{\nu_{zx}}{E_z} & 0 & 0 & 0 \\ -\dfrac{\nu_{xy}}{E_x} & \dfrac{1}{E_y} & -\dfrac{\nu_{zy}}{E_z} & 0 & 0 & 0 \\ -\dfrac{\nu_{xz}}{E_x} & -\dfrac{\nu_{yz}}{E_y} & \dfrac{1}{E_z} & 0 & 0 & 0 \\ 0 & 0 & 0 & \dfrac{1}{G_{yz}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \dfrac{1}{G_{xz}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \dfrac{1}{G_{xy}} \end{bmatrix} \tag{1.104}$$

Since the compliance must be symmetric, it must satisfy (1.93) with $i, j = x, y, z$. Therefore, it is possible to compute the apparent engineering properties of a laminate in terms of the laminate compliance as follows

$$\begin{aligned} E_x &= 1/S_{11} & \nu_{xy} &= -S_{21}/S_{11} \\ E_y &= 1/S_{22} & \nu_{xz} &= -S_{31}/S_{11} \\ E_z &= 1/S_{33} & \nu_{yz} &= -S_{32}/S_{22} \\ G_{yz} &= 1/S_{44} \\ G_{xz} &= 1/S_{55} \\ G_{xy} &= 1/S_{66} \end{aligned} \tag{1.105}$$

Example 1.6 *Compute the laminate properties of* $[0/90/\pm 30]_S$ *with* $t_k = 1.5$ *mm,* $E_f = 241$ *GPa,* $\nu_f = 0.2$*,* $E_m = 3.12$ *GPa,* $\nu_m = 0.38$*, fiber volume fraction* $V_f = 0.6$*, where f,m, denote fiber and matrix, respectively.*

Solution to Example 1.6 *First use periodic microstructure micromechanics (6.8) to obtain the layer properties (in MPa).*

$$\begin{array}{lll} E_1 = 145,880 & G_{12} = 4,386 & \nu_{12} = \nu_{13} = 0.263 \\ E_2 = 13,312 & G_{23} = 4,528 & \nu_{23} = 0.470 \end{array}$$

Then, compute the compliance matrix $[S']$ *using (1.91), the rotation matrix* $[T]$ *using (1.34), the compliance* $[S]$ *in laminate coordinate system using (1.53), and the stiffness* $[C] = [S]^{-1}$ *in laminate coordinate system for each layer. Then, average them using (1.102), invert the average, and finally using (1.105) get*

$$\begin{array}{lll} E_x = 78,901 & G_{xy} = 17,114 & \nu_{xy} = 0.320 \\ E_y = 47,604 & G_{yz} = 4,475 & \nu_{yz} = 0.364 \\ E_z = 16,023 & G_{xz} = 4,439 & \nu_{xz} = 0.280 \end{array}$$

Suggested Problems

Problem 1.1 *Using the principle of virtual work (PVW), find a quadratic displacement function $u(x)$ in $0 < x < L$ of a tapered slender rod of length L, fixed at the origin and loaded axially in tension at the free end. The cross-section area changes lineally and the areas are $A_1 > A_2$ at the fixed and free ends, respectively. The material is homogeneous and isotropic with modulus E.*

Problem 1.2 *Using the principle of virtual work (PVW), find a quadratic rotation angle function $\theta(x)$ in $0 < x < L$ of a tapered slender shaft of circular cross section and length L, fixed at the origin and loaded by a torque T at the free end. The cross-section area changes lineally and the areas are $A_1 > A_2$ at the fixed and free ends, respectively. The material is homogeneous and isotropic with shear modulus G.*

Problem 1.3 *Construct a rotation matrix $[a]$ resulting from three consecutive reflections about (a) the x-y plane, (b) the x-z plane, (c) the y-z plane. The resulting system does not follow the right-hand rule.*

Problem 1.4 *Construct three rotation matrices $[a]$ for rotations $\theta = \pi$ about (a) the x-axis, (b) the y-axis, (c) the z-axis.*

Problem 1.5 *Using*

$$\sigma = \begin{bmatrix} 10 & 2 & 1 \\ 2 & 5 & 1 \\ 1 & 1 & 3 \end{bmatrix}$$

and $[a]$ of Ex. 1.2, verify that (1.29) yields the same result as (1.26).

Problem 1.6 *Write a computer program to evaluate the compliance and stiffness matrices in terms of engineering properties. Take the input from a file and the output to another file. Validate the program with your own examples. You may use material properties from [1, Tables 1.3–1.4] and assume the material is transversely isotropic as per Section 1.12.4. Show all work in a report.*

Problem 1.7 *Write a computer program to transform the stiffness and compliance matrix from lamina coordinates C', S', to another coordinate system C, S, by a rotation $-\theta$ around the z-axis (Figure 1.7). The data C', S', θ, should be read from a file. The output C, S should be written to another file. Validate your program with your own examples. You may use material properties from [1, Tables 1.3–1.4] and assume the material is transversely isotropic as per Section 1.12.4. Show all work in a report.*

Problem 1.8 *Verify numerically (1.92) against $[S]^{-1}$ for the material of your choice. You may use material properties from [1, Tables 1.3–1.4] and assume the material is transversely isotropic as per Section 1.12.4.*

Problem 1.9 *The following data has been obtained experimentally for a composite based on a unidirectional carbon-epoxy prepreg (MR50 carbon fiber at 63% by volume in LTM25 Epoxy). Determine if the restrictions on elastic constants are satisfied.*

$$
\begin{aligned}
E_1 &= 156.403\ GPa, & E_2 &= 7.786\ GPa \\
\nu_{12} &= 0.352, & \nu_{21} &= 0.016 \\
G_{12} &= 3.762\ GPa \\
\sigma^u_{1t} &= 1.826\ GPa, & \sigma^u_{1c} &= 1.134\ GPa \\
\sigma^u_{2t} &= 19\ MPa, & \sigma^u_{2c} &= 131\ MPa \\
\sigma^u_{6} &= 75\ MPa \\
\epsilon^u_{1t} &= 11,900\ 10^{-6}, & \epsilon^u_{1c} &= 8,180\ 10^{-6} \\
\epsilon^u_{2t} &= 2,480\ 10^{-6}, & \epsilon^u_{2c} &= 22,100\ 10^{-6} \\
\gamma^u_{12} &= 20,000\ 10^{-6}
\end{aligned}
$$

Problem 1.10 *Explain contracted notation for stresses and strains.*

Problem 1.11 *What is an orthotropic material and how many constants are needed to describe it?*

Problem 1.12 *What is a transversely isotropic material and how many constants are needed to describe it?*

Problem 1.13 *Use the three rotation matrices in Problem 1.4 to verify (1.48) numerically.*

Problem 1.14 *Prove (1.73) using (1.71) and (1.91).*

Problem 1.15 *Demonstrate that a material having two perpendicular planes of symmetry also has a third. Apply a reflection about the 2-3 plane to (1.68) using the procedure in Section 1.12.3.*

Problem 1.16 *What is a plane stress assumption?*

Problem 1.17 *Write a computer program to evaluate the laminate engineering properties for symmetric balanced laminates. All layers are of the same material. Input data consist of all the engineering constants for a transversely isotropic material, number of layers N, thickness and angle for all the layers t_k, θ_k with $k = 1...N$. Use Section 1.15, 1.12.4, and 1.13.*

Chapter 2

Introduction to Finite Element Analysis

In this textbook, the finite element method (FEM) is used as a tool to solve practical problems. For the most part, commercial packages, mainly *Abaqus*, are used in the examples. Computer programming is limited to implementing material models and post processing algorithms. When commercial codes lack needed features, other codes are used, which are provided in [5]. A basic understanding of the finite element method is necessary for effective use of any finite element software. Therefore, this chapter contains a brief introduction intended for those readers who have not had a formal course or prior knowledge about the finite element method. Furthermore, an introduction to the *Abaqus*/CAE graphical user interface (GUI) is presented to familiarize the reader with the typical procedures used for finite element modeling using any commercial software.

2.1 Basic FEM Procedure

Consider the axial deformation of a rod. The ordinary differential equation (ODE) describing the deformation of the rod is

$$-\frac{d}{dx}\left(EA\frac{du}{dx}\right) - f = 0 \quad ; \quad 0 \leq x \leq L \tag{2.1}$$

where E, A are the modulus and cross-section area of the rod, respectively, and f is the distributed force. The boundary conditions for the case illustrated in Figure 2.1 are

$$\begin{aligned} u(0) &= 0 \\ \left[\left(EA\frac{du}{dx}\right)\right]_{x=L} &= P \end{aligned} \tag{2.2}$$

As it is customary in mechanics of materials textbooks, the real rod shown in Figure 2.1(a) is mathematically modeled as a line in Figure 2.1(b). The rod occupies the domain $[0, L]$ along the real axis x.

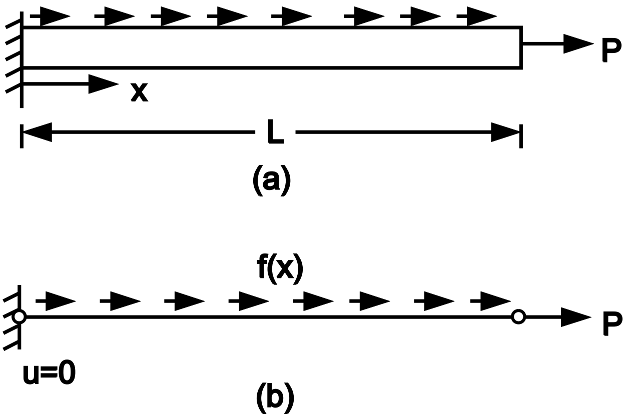


Fig. 2.1: Physical and mathematical (idealization) model.

2.1.1 Discretization

The next step is to divide the domain into discrete elements, as shown in Figure 2.2.

2.1.2 Element Equations

To derive the element equations, an integral form of the ordinary differential equation (ODE) is used, which is obtained by integrating the product of the ODE times a weight function v as follows

$$0 = \int_{x_A}^{x_B} v \left[-\frac{d}{dx}\left(EA\frac{du}{dx} \right) - f \right] dx \tag{2.3}$$

This is called a weak form because the solution $u(x)$ does not have to satisfy the ODE (2.1) for all and every one of the infinite values of x in $[0, L]$, in a strong sense. Instead, the solution $u(x)$ only has to satisfy the ODE in (2.3) in a weighted average sense. It is therefore easier to find a weak solution than a strong one. Although for the case of the rod, the strong (exact) solution is known, most problems of composite mechanics do not have exact solutions. The governing equation is obtained by integrating (2.3) by parts as follows

$$0 = \int_{x_A}^{x_B} EA\frac{dv}{dx}\frac{du}{dx}dx - \int_{x_A}^{x_B} vf dx - \left[v\left(EA\frac{du}{dx} \right) \right]_{x_A}^{x_B} \tag{2.4}$$

where $v(x)$ is a weight function, which is usually set equal to the primary variable $u(x)$. From the boundary term, it is concluded that

– specifying $v(x)$ at x_A *or* x_B is an essential boundary condition

– specifying $\left(EA\frac{du}{dx} \right)$ at either end is the natural boundary condition

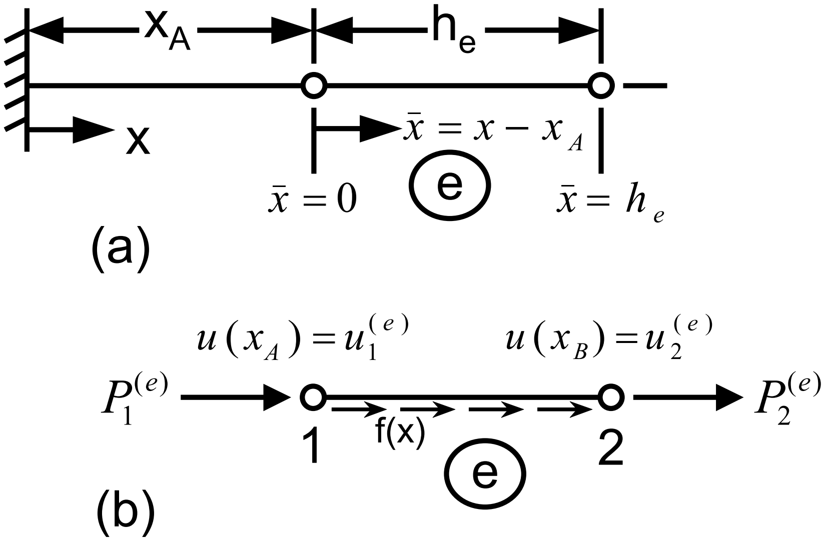


Fig. 2.2: Discretization into elements.

While $u(x)$ is the *primary variable*, $\left(EA\frac{du}{dx}\right) = EA\epsilon_x = A\sigma_x$ is the *secondary variable*. Let

$$\begin{aligned} u(x_A) &= u_1^e \\ u(x_B) &= u_2^e \\ -\left[\left(EA\frac{du}{dx}\right)\right]_{x_A} &= P_1^e \\ \left[\left(EA\frac{du}{dx}\right)\right]_{x_B} &= P_2^e \end{aligned} \tag{2.5}$$

Then, the governing equation becomes

$$0 = \int_{x_A}^{x_B} \left(EA\frac{dv}{dx}\frac{du}{dx} - vf\right) dx - P_1^e v(x_A) - P_2^e v(x_B) = B(v,u) - l(v) \tag{2.6}$$

with

$$\begin{aligned} B(u,v) &= \int_{x_A}^{x_B} EA\frac{dv}{dx}\frac{du}{dx} dx \\ l(v) &= \int_{x_A}^{x_B} vf dx + P_1^e v(x_A) + P_2^e v(x_B) \end{aligned} \tag{2.7}$$

2.1.3 Approximation over an Element

Now, the unknown $u(x)$ is approximated as a linear combination (series expansion) of known functions $N_i^e(x)$ and unknown coefficients a_j^e, as

$$u_e(x) = \sum_{j=1}^{n} a_j^e N_j^e(x)$$

where a_j^e are the coefficients to be found and $N_j^e(x)$ are the interpolation functions. For the weight function $v(x)$, the Ritz method can be used [4], in which $v(x) = N_j^e(x)$. Substituting in the governing equation (2.6) we get

$$\sum_{j=1}^{n} \left(\int_{x_A}^{x_B} EA \frac{dN_i^e}{dx} \frac{dN_j^e}{dx} dx \right) a_j^e = \int_{x_A}^{x_B} N_i^e f dx + P_1^e N_i^e(x_A) + P_2^e N_i^e(x_B) \tag{2.8}$$

which can be written as

$$\sum_{j=1}^{n} K_{ij}^e a_j^e = F_i^e \tag{2.9}$$

or in matrix form

$$[K^e]\{a^e\} = \{F^e\} \tag{2.10}$$

where $[K^e]$ is the element stiffness matrix, $\{F^e\}$ is the element vector equivalent force, and $\{a^e\}$ are the element unknown parameters.

2.1.4 Interpolation Functions

Although any complete set of linearly independent functions could be used as interpolation functions, it is convenient to choose the function in such a way that the unknown coefficients represent the nodal displacements, that is $a_i = u_i$. For a two-node element spanning the interval $x_e \leq x \leq x_{e+1}$, the following linear interpolation functions (Figure 2.3) can be used

$$\begin{aligned} N_1^e &= \frac{x_{e+1} - x}{h_e} \\ N_2^e &= \frac{x - x_e}{h_e} \end{aligned} \tag{2.11}$$

where $h_e = x_{e+1} - x_e$ is the element length. These interpolation functions satisfy the following conditions

$$N_i^e(x_j) = \begin{Bmatrix} 0 & if & i \neq j \\ 1 & if & i = j \end{Bmatrix} \tag{2.12}$$

$$\sum_{i=1}^{2} N_i^e(x) = 1 \tag{2.13}$$

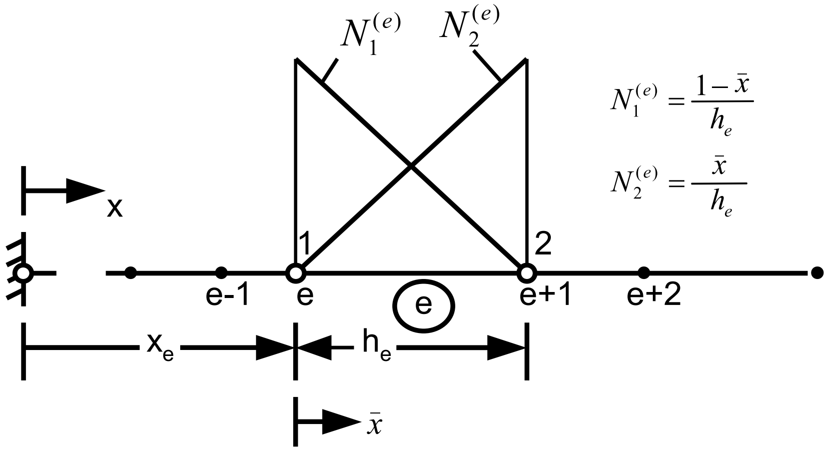

$N_1^{(e)}$
$N_2^{(e)}$
$N_1^{(e)} = \frac{1-\bar{x}}{h_e}$
$N_2^{(e)} = \frac{\bar{x}}{h_e}$
x
1
2
e-1
e
e
e+1
e+2
x_e
h_e
$\bar{x}$

Fig. 2.3: Linear interpolation functions for a two-node element rod.

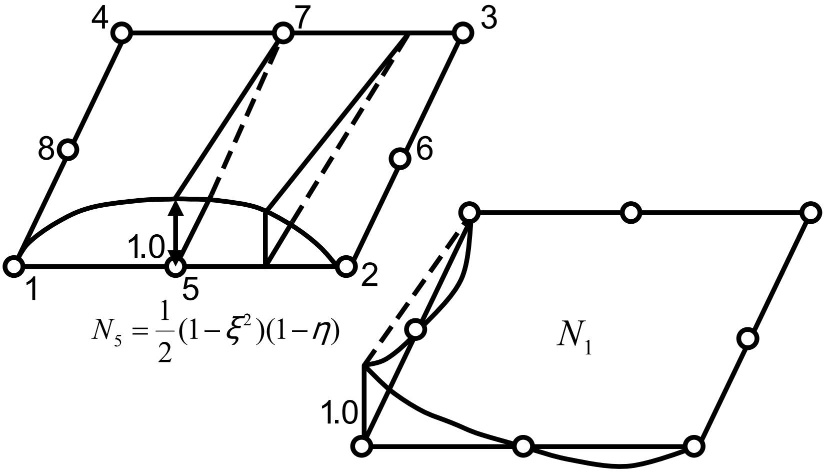


Fig. 2.4: Two-dimensional interpolation functions.

which guarantees that the unknown coefficients represent the nodal displacements, i.e., $a_i = u_i$.

Many other interpolation functions can be used, each one with some advantages and disadvantages. The interpolation functions are intimately related to the number of nodes of the element. Figure 2.4 illustrates the shape of the interpolation functions N_1 and N_5 (corresponding to nodes 1 and 5) in an eight-node shell element.

Broadly speaking, more nodes per element imply more accuracy and less need for a fine mesh, but also imply higher cost in terms of computer time. Figure 2.5 illustrates how the approximate solution converges to the exact one as the number of elements increases from 2 to 4 or as the number of nodes in the element increases from 2 for the linear element to 3 for the quadratic element.

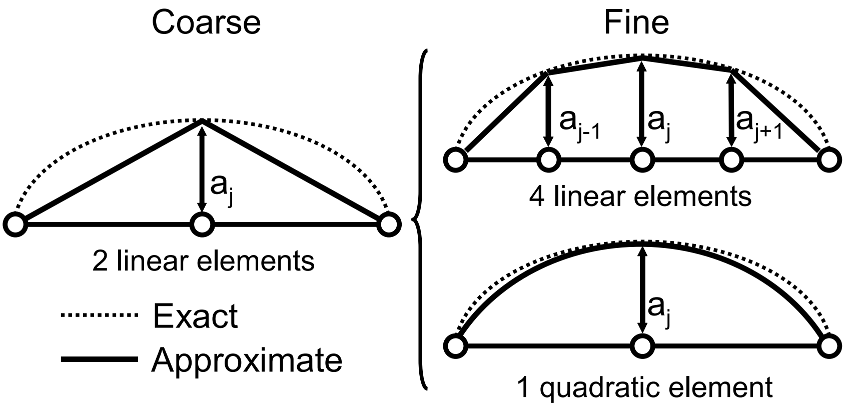


Fig. 2.5: Discretization error.

2.1.5 Element Equations for a Specific Problem

With interpolation functions that satisfy the conditions in (2.12-2.13), it is possible to rewrite (2.10) as

$$[K^e]\{u^e\} = \{F^e\} \tag{2.14}$$

where $\{u^e\}$ are the nodal displacements, $[K^e]$ is the element stiffness matrix given by

$$[K^e] = \left[\begin{array}{cc} \int_{x_A}^{x_B} EA\frac{dN_1^e}{dx}\frac{dN_1^e}{dx}dx & \int_{x_A}^{x_B} EA\frac{dN_1^e}{dx}\frac{dN_2^e}{dx}dx \\ \int_{x_A}^{x_B} EA\frac{dN_2^e}{dx}\frac{dN_1^e}{dx}dx & \int_{x_A}^{x_B} EA\frac{dN_2^e}{dx}\frac{dN_2^e}{dx}dx \end{array} \right] \tag{2.15}$$

and $\{F^e\}$ is the element force vector

$$\{F_i^e\} = \left\{ \begin{array}{c} \int_{x_A}^{x_B} N_1^e f dx + P_1^e \\ \int_{x_A}^{x_B} N_2^e f dx + P_2^e \end{array} \right\} \tag{2.16}$$

For a two-node rod element number e, the constant cross-section area A_e, the element length h_e, and the modulus E are fixed. These values define the tensile-compression element stiffness as

$$k^e = \frac{EA_e}{h_e} \tag{2.17}$$

The external loads on the element are the distributed force f_e, the force at end number 1, P_1^e, and the force at end number 2, P_2^e. Using these values, the linear interpolation functions (2.11), as well as (2.15) and (2.16), the element matrix stiffness and the equivalent nodal forces become

$$[K^e] = \left[\begin{array}{cc} k^e & -k^e \\ -k^e & k^e \end{array} \right] = \frac{EA_e}{h_e} \left[\begin{array}{cc} 1 & -1 \\ -1 & 1 \end{array} \right] \tag{2.18}$$

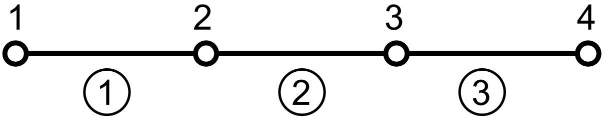


Fig. 2.6: Connectivity between three two-node elements.

$$\{F^e\} = \frac{f_e h_e}{2} \left\{ \begin{array}{c} 1 \\ 1 \end{array} \right\} + \left\{ \begin{array}{c} P_1^e \\ P_2^e \end{array} \right\} \tag{2.19}$$

2.1.6 Assembly of Element Equations

The element unknown parameters correspond to displacements at the element nodes. Since a node must have the same displacement on both adjacent elements, the value is unique. For example, using the connectivity of elements shown in Figure 2.6, unique labels are assigned to the displacements, using capital letters. While a superscript denotes an element number, a subscript indicates a nodal number, as follows

$$\begin{aligned} u_1^1 &= U_1 \\ u_2^1 &= U_2 = u_1^2 \\ u_1^2 &= U_3 = u_1^3 \\ u_2^3 &= U_4 \end{aligned} \tag{2.20}$$

Now, the element equations can be assembled into the global system. First, the contribution of element #1 is

$$\begin{bmatrix} k^1 & -k^1 & 0 & 0 \\ -k^1 & k^1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \left\{ \begin{array}{c} U_1 \\ U_2 \\ U_3 \\ U_4 \end{array} \right\} = \left\{ \begin{array}{c} f_1 h_1/2 \\ f_1 h_1/2 \\ 0 \\ 0 \end{array} \right\} + \left\{ \begin{array}{c} P_1^1 \\ P_2^1 \\ 0 \\ 0 \end{array} \right\} \tag{2.21}$$

Add the contribution of element #2, as follows

$$\begin{bmatrix} k^1 & -k^1 & 0 & 0 \\ -k^1 & k^1 + k^2 & -k^2 & 0 \\ 0 & -k^2 & k^2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \left\{ \begin{array}{c} U_1 \\ U_2 \\ U_3 \\ U_4 \end{array} \right\} = \left\{ \begin{array}{c} f_1 h_1/2 \\ f_1 h_1/2 + f_2 h_2/2 \\ f_2 h_2/2 \\ 0 \end{array} \right\} + \left\{ \begin{array}{c} P_1^1 \\ P_2^1 + P_1^2 \\ P_2^2 \\ 0 \end{array} \right\} \tag{2.22}$$

Finally, add element #3 to obtain the fully assembled system, as follows

$$\begin{bmatrix} k^1 & -k^1 & 0 & 0 \\ -k^1 & k^1+k^2 & -k^2 & 0 \\ 0 & -k^2 & k^2+k^3 & -k^3 \\ 0 & 0 & -k^3 & k^3 \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{Bmatrix} = \frac{1}{2} \begin{Bmatrix} f_1h_1 \\ f_1h_1+f_2h_2 \\ f_2h_2+f_3h_3 \\ f_3h_3 \end{Bmatrix} + \begin{Bmatrix} P_1^1 \\ P_2^1+P_1^2 \\ P_2^2+P_1^3 \\ P_2^3 \end{Bmatrix} \tag{2.23}$$

2.1.7 Boundary Conditions

By equilibrium (see Figure 2.2), the internal loads cancel whenever two elements share a node, or

$$\begin{aligned} P_2^1 + P_1^2 &= 0 \\ P_2^2 + P_1^3 &= 0 \end{aligned} \tag{2.24}$$

The remaining P_1^1 and P_3^2 are the forces at the end of the rod. If either end of the rod is fixed, then the displacement must be set to zero at that end. Say the end at $x = 0$ is fixed, then $U_1 = 0$. If the end at $x = L$ is free, then P_3^2 must be specified, since $U_4 \neq 0$. If it is not specified, then it is assumed that the force is zero.

2.1.8 Solution of the Equations

Since $U_1 = 0$, eliminating the first row and column of the stiffness matrix, a 3 × 3 system of algebraic equations is obtained, and solved for 3 unknowns: U_2, U_3, U_4. Once a solution for U_2 is found, the reaction P_1^1 is computed from the first equation of (2.23), as follows

$$-k^1U_2 = \frac{f_1h_1}{2} + P_1^1 \tag{2.25}$$

2.1.9 Solution Inside the Elements

Now, the solution U_i at 4 points along the rod is available. Next, the solution at any location x can be computed by interpolating with the interpolation functions, as follows

$$U^e(x) = \sum_{j=1}^{2} U_j^e N_j^e(x) \tag{2.26}$$

or

$$u(x) = \begin{cases} U_1N_1^1(x) + U_2N_2^1(x) & if \quad 0 \leq x \leq h_1 \\ U_2N_1^2(x) + U_3N_2^2(x) & if \quad h_1 \leq x \leq h_1 + h_2 \\ U_3N_1^3(x) + U_4N_2^3(x) & if \quad h_1 + h_2 \leq x \leq h_1 + h_2 + h_3 \end{cases} \tag{2.27}$$

2.1.10 Derived Results

Strains

Strains are computed using (1.5) directly from the known displacements inside the element. For example,

$$\epsilon_x = \frac{du}{dx} = \sum_{j=1}^{2} U_j^e \frac{dN_j^e}{dx} \tag{2.28}$$

Note that if $N_j^e(x)$ are linear functions, the strains are constant over the element. In general the quality of the strains is one order of magnitude poorer than the primary variable (displacements).

Stresses

Stress values are usually computed from strains through the constitutive equations. In this example, with one-dimensional stress-strain behavior

$$\sigma_x = E\ \epsilon_x \tag{2.29}$$

Note that the quality of stresses is the same as that of the strains.

2.2 General Finite Element Procedure

The derivation of the element equations, assembly, and solution for any type of elements is similar to that of the one-dimensional rod element described in Section 2.1, with the exception that the principle of virtual work (PVW, 1.16) is used instead of the governing equation (2.1). The PVW provides a weak form similar to that in (2.4). Expanding (1.16) for full 3D state of deformation, the internal virtual work is

$$\begin{aligned}\delta W_I &= \int (\sigma_{xx}\delta\epsilon_{xx} + \sigma_{yy}\delta\epsilon_{yy} + \sigma_{zz}\delta\epsilon_{zz} + \sigma_{yz}\delta\gamma_{yz} + \sigma_{xz}\delta\gamma_{xz} + \sigma_{xy}\delta\gamma_{xy})\, dV \\ &= \int_V \underline{\sigma}^T \underline{\delta\epsilon}\, dV \end{aligned} \tag{2.30}$$

where

$$\begin{aligned}\underline{\sigma}^T &= \{\sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \sigma_{yz}, \sigma_{xz}, \sigma_{xy}\} \\ \delta\underline{\epsilon}^T &= \{\delta\epsilon_{xx}, \delta\epsilon_{yy}, \delta\epsilon_{zz}, \delta\gamma_{yz}, \delta\gamma_{xz}, \delta\gamma_{xy}\}\end{aligned} \tag{2.31}$$

Next, the external work is

$$\delta W_E = \int_V \underline{f}^T \underline{\delta u}\, dV + \int_S \underline{t}^T \underline{\delta u}\, dS \tag{2.32}$$

where the volume forces per unit volume and surface forces per unit area are

$$\begin{aligned} \underline{f}^T &= \{f_x, f_y, f_z\} \\ \underline{t}^T &= \{t_x, t_{y,} t_z\} \end{aligned} \tag{2.33}$$

Here, underline (_) denotes a one-dimensional array, not necessarily a vector. For example, $\underline{u}$ is a vector but $\underline{\sigma}$ are the six components of stress arranged in a six-element array. The virtual strains are strains that would be produced by virtual displacements $\underline{\delta u}\ (\underline{x})$. Therefore, virtual strains are computed from virtual displacements using the strain-displacement equations (1.5). In matrix notation

$$\begin{aligned} \underline{\epsilon} &= \underline{\underline{\partial}}\ \underline{u} \\ \underline{\delta\epsilon} &= \underline{\underline{\partial}}\ \underline{\delta u} \end{aligned} \tag{2.34}$$

where

$$\underline{\underline{\partial}} = \begin{bmatrix} \dfrac{\partial}{\partial x} & 0 & 0 & \dfrac{\partial}{\partial y} & 0 & \dfrac{\partial}{\partial z} \\ 0 & \dfrac{\partial}{\partial y} & 0 & \dfrac{\partial}{\partial x} & \dfrac{\partial}{\partial z} & 0 \\ 0 & 0 & \dfrac{\partial}{\partial z} & 0 & \dfrac{\partial}{\partial y} & \dfrac{\partial}{\partial x} \end{bmatrix}^T \tag{2.35}$$

Then, the PVW is written in matrix notation as

$$\int_V \underline{\sigma}^T \underline{\underline{\partial}}\ \underline{\delta u}\ dV = \int_V \underline{f}^T \underline{\delta u}\ dV + \int_S \underline{t}^T \underline{\delta u}\ dS \tag{2.36}$$

The integrals over the volume V and surface S of the body can be broken element by element over m elements, as

$$\sum_{e=1}^{m} \left[\int_{V_e} \underline{\sigma}^T \underline{\underline{\partial}}\ \underline{\delta u}\ dV \right] = \sum_{e=1}^{m} \left[\int_{V_e} \underline{f}^T \underline{\delta u}\ dV + \int_{S_e} \underline{t}^T \underline{\delta u}\ dS \right] \tag{2.37}$$

Whenever two elements share a surface, the contributions of the second integral cancel out, just as the internal loads canceled in Section 2.1.7. The stress components are given by the constitutive equations. For a linear material

$$\underline{\sigma} = \underline{\underline{C}}\ \underline{\epsilon} \tag{2.38}$$

with $\underline{\underline{C}}$ given by (1.68). The internal virtual work over each element becomes

$$\delta W_I^e = \int_{V_e} \underline{\sigma}^T \underline{\delta\epsilon}\ dV = \int_{V_e} \underline{\epsilon}^T \underline{\underline{C}}\ \underline{\delta\epsilon}\ dV \tag{2.39}$$

The expansion of the displacements can be written in matrix form as

$$\underline{u} = \underline{\underline{N}}\ \underline{a} \tag{2.40}$$

where $\underline{N}$ contains the element interpolation functions and $\underline{a}$ the nodal displacements of the element, just as in Section 2.1.4. Therefore, the strains are

$$\underline{\epsilon} = \underline{\underline{\partial}}\,\underline{u} = \underline{\underline{\partial}}\,\underline{N}\,\underline{a} = \underline{\underline{B}}\,\underline{a} \tag{2.41}$$

where $\underline{\underline{B}} = \underline{\underline{\partial}}\,\underline{N}$ is the strain-displacement matrix. Now, the discretized form of the internal virtual work over an element can be computed as

$$\delta W_I^e = \int_{V_e} \underline{a}^T \underline{\underline{B}}^T \underline{\underline{C}}\,\underline{\underline{B}}\,\delta\underline{a}\,dV = \underline{a}^T \int_{V_e} \underline{\underline{B}}^T \underline{\underline{C}}\,\underline{\underline{B}}\,dV \delta\underline{a} = \underline{a}^T \underline{\underline{K}}^e\,\delta\underline{a} \tag{2.42}$$

where the element stiffness matrix K^e is

$$\underline{\underline{K}}^e = \int_{V_e} \underline{\underline{B}}^T \underline{\underline{C}}\,\underline{\underline{B}}\,dV \tag{2.43}$$

The external virtual work becomes

$$\begin{aligned} \delta W_E^e &= \int_{V_e} \underline{f}^T \delta\underline{u}\,dV + \int_{S_e} \underline{t}^T \delta\underline{u}\,dS \\ &= \left(\int_{V_e} \underline{f}^T \underline{N}\,dV + \int_{S_e} \underline{t}^T \underline{N}\,dS \right) \delta\underline{a} = \left(\underline{P}^e\right)^T \delta\underline{a} \end{aligned} \tag{2.44}$$

where the element force vector is

$$\underline{P}^e = \int_{V_e} \underline{N}^T \underline{f}\,dV + \int_{S_e} \underline{N}^T \underline{t}\,dS \tag{2.45}$$

The integrals over the element volume V_e and element surface S_e are usually evaluated numerically by the Gauss integration procedure. For the volume integral, such a procedure needs evaluation of the integrand at a few points inside the volume. Such points, which are called Gauss points, are important for two reasons. First, the constitutive matrix C is evaluated at those locations. Second, the most accurate values of strains (and stresses) are obtained at those locations too.

The assembly of the element equations δW_I^e and δW_E^e into the PVW for the whole body is done similarly to the process in Section 2.1.6. Obviously the process is more complicated than for rod elements. The details of such process and its computer programming are part of finite element technology, which is outside the scope of this textbook. Eventually all the element stiffness matrices K^e and element force vectors P^e are assembled into a global system for the whole body

$$\underline{\underline{K}}\,\underline{a} = \underline{P} \tag{2.46}$$

Next, boundary conditions are applied on the system (2.46) in a systematic way resembling the procedure in Section 2.1.6. Next, the algebraic system of equations (2.46) is solved to find the nodal displacement array $\underline{a}$ over the whole body. Since the nodal displacements results for every element can be found somewhere in $\underline{a}$, it is possible to go back to (2.34) and to (2.38) to compute the strains and stresses anywhere inside the elements.

Example 2.1 *Compute the element stiffness matrix (2.43) and the equivalent force vector (2.45) of a rod discretized with one element. Use linear interpolation functions such as (2.11). Compare the result with (2.18-2.19).*

Solution to Example 2.1 *Let A_e be the transverse area of the rod and h_e the element length, with $x_e = 0$ and $x_{e+1} = h_e$. Substituting these values in the linear interpolation functions from equation (2.11), the interpolation functions arrays are obtained as follows*

$$\underline{N^T} = \begin{bmatrix} N_1^e \\ N_2^e \end{bmatrix} = \begin{bmatrix} \dfrac{x_{e+1} - x}{h_e} \\ \dfrac{x - x_e}{h_e} \end{bmatrix} = \begin{bmatrix} 1 - x/h_e \\ x/h_e \end{bmatrix}$$

The strain-displacement array is obtained as

$$\underline{\underline{B^T}} = \underline{\underline{\partial}}\ \underline{N^T} = \begin{bmatrix} \partial N_1^e/\partial x \\ \partial N_2^e/\partial x \end{bmatrix} = \begin{bmatrix} -1/h_e \\ 1/h_e \end{bmatrix}$$

The rod element has a one-dimensional strain-stress state with linear elastic behavior. Therefore

$$\underline{\underline{C}} = E$$

Then, using equation (2.43) we can write

$$\underline{\underline{K^e}} = \int_{V_e} \underline{\underline{B}}^T \underline{\underline{C}}\ \underline{\underline{B}}\, dV = \int_0^{h_e} \begin{bmatrix} -1/h_e & 1/h_e \end{bmatrix} E \begin{bmatrix} -1/h_e \\ 1/h_e \end{bmatrix} A_e dx$$

The element stiffness matrix is obtained by integration

$$[K^e] = \frac{EA_e}{h_e} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

To calculate the equivalent vector force, f_e is defined as the distributed force on element, P_1^e is the force at end $x = 0$, and P_2^e is the force at end $x = h_e$. Substituting into equation (2.45) we obtain

$$\underline{P^e} = \int_{V_e} \underline{N}^T \underline{f}\, dV + \int_{S_e} \underline{N}^T \underline{t}\, dS = \int_0^{h_e} \begin{bmatrix} 1 - x/h_e \\ x/h_e \end{bmatrix} f_e dx + \begin{bmatrix} P_1^e \\ P_2^e \end{bmatrix}$$

The element equivalent force vector is obtained by integration

$$\underline{P^e} = \frac{f_e h_e}{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} P_1^e \\ P_2^e \end{bmatrix}$$

2.3 Solid Modeling, Analysis, and Visualization

Many commercial programs exist with finite element analysis capabilities for different engineering disciplines. They help solve a variety of problems from simple linear static analysis to nonlinear transient analysis. A few of these codes, such as *ANSYS* and *Abaqus*, have special capabilities to analyze composite materials and they accept custom, user-programmed constitutive equations and element formulations. Since these software packages not only provide analysis tools, geometric modeling, and visualization of results, but also they can be integrated in the larger design, production, and product life-cycle process, they are often called complete analysis environments or computer aided engineering (CAE) systems.

Modern FEA software are commonly organized into three blocks: the pre-processor, the processor, and the post-processor. In the pre-processor, the model is built defining the geometry, material properties, and element type. Also, loads and boundary conditions are entered in the pre-processor, but they may also be entered during the solution phase. With this information, the processor can compute the stiffness matrix and the force vector. Next, the algebraic equations (2.46) are solved and the solution is obtained in the form of displacement values. In the last block–the post-processor–derived results, such as stress, strain, and failure ratios, are computed. The solution can be reviewed using graphic tools.

In the remainder of this chapter, a general description of the procedures and the specific steps for a basic finite element analysis (FEA) are presented using examples executed with *Abaqus* [7]. Although the emphasis of this textbook is on mechanics of composite materials, concepts are illustrated with examples that are solved using *Abaqus*. Solutions to similar examples are available in [8], using *ANSYS* [9].

The first requirement of the model is the geometry. Then, material properties are given for the various parts that make up the geometry. Next, loads and boundary conditions are applied on the geometry. Next, the geometry is discretized into elements, which are defined in terms of the nodes and element connectivity. The element type is chosen to represent the type of problem to be solved. Next, the model is solved. Finally, derived results are computed and visualized.

2.3.1 Model Geometry

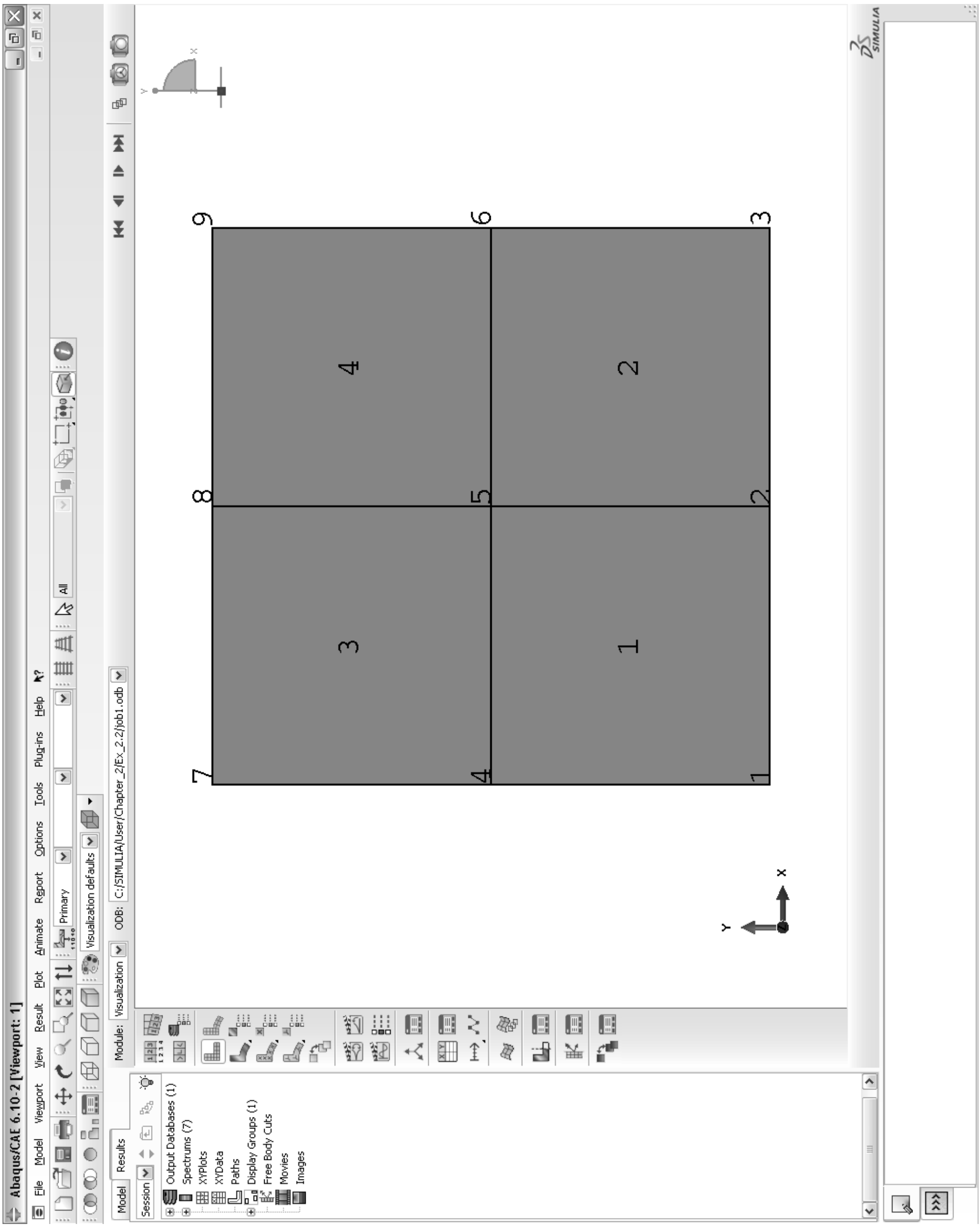
The model geometry is obtained specifying all nodes, their position, and the element connectivity. The connectivity information allows the program to assemble the element stiffness matrix and the element equivalent force vector to obtain the global equilibrium equations, as shown in Section 2.1.6.

There are two ways to generate the model. The first is to manually create a mesh. The second is to use solid modeling, and then mesh the solid to get the node and element distribution.

Manual Meshing

Manual mesh generation was the only method available before solid modeling became widespread among commercial packages. It is still the only option with some older and custom software, although in those cases it is always possible to use a general purpose, solid modeling pre-processor to generate the mesh. With manual meshing, the user creates nodes, then connects the nodes into elements. Afterward, the user applies boundary conditions and loads directly on nodes and/or elements. Manual meshing is used in Example 2.2.

Example 2.2 *Use manual meshing to set up a finite element model of a two-dimensional square sheet, in a state of plane stress, with dimensions* 20×20 *mm* $\times$ 4 *mm and mesh it with a* 2×2 *mesh of continuum, plane stress, square, CPS4R elements in* Abaqus. *The*



Abaqus/CAE 6.10-2 [Viewport: 1]

Module: Visualization

ODB: C:/SIMULIA/User/Chapter_2/Ex_2.2/Job1.odb

Fig. 2.7: Square mesh in Ex. 2.2.

left edge (nodes 1, 4, 7 in Fig. 2.7) should be horizontally restrained. Also, node 1 should be vertically restrained. A uniform surface load of $-9.5\ MN/mm^2$ *should be applied to the right edge (nodes 3, 6, 9). The material is steel with* $E = 210,000\ MPa, \nu = 0.3$*. Run the model from the DOS shell window. Then, use* Abaqus*/CAE to visualize the results.*

Solution to Example 2.2 *The commands listed below define the model geometry by using manual meshing, and they are written into what is called an input file (*`Ex_2.2.inp`*). They are available on the Web site [5, file name: Ex_2.2.inp].*

The `*Node` *command line is followed by a list of nodes, each line containing the node number, plus* x, y*, and possibly* z *coordinates of each node.*

The `*Element` *command line is followed by a list of elements, each line containing the element number, and a sequence of node numbers going counterclockwise (ccw) around the boundary of the element (see Fig. 2.7).*

The meaning of the remaining command lines is either obvious or can be researched in the Abaqus *Keywords Reference Manual [10].*

```
*Heading
Example 2.2 FEA of Composite Materials: using Abaqus
*Part, name=Part1
*Node
  1,          0.,           0.
  2,         10.,           0.
  3,         20.,           0.
  4,          0.,          10.
  5,         10.,          10.
  6,         20.,          10.
  7,          0.,          20.
  8,         10.,          20.
  9,         20.,          20.
*Element, type=CPS4R
  1,  1,  2,  5,  4
  2,  2,  3,  6,  5
  3,  4,  5,  8,  7
  4,  5,  6,  9,  8
*Nset, nset=NamedSet, generate
  1,   9,   1
*Elset, elset=NamedSet, generate
  1,   4,   1
** Section: Section-1
*Solid Section, elset=NamedSet, material=Material-1
4.,
*End Part
*Assembly, name=Assembly
*Instance, name=Part1-1, part=Part1
*End Instance
*Nset, nset=Nodeset1, internal, instance=Part1-1
 1
*Nset, nset=Nodeset2, internal, instance=Part1-1
 4,  7
*Elset, elset=Elset1, internal, instance=Part1-1
  2,   4
```

```
*Surface, type=ELEMENT, name=Surface1, internal
Elset1, S2
*End Assembly
*Material, name=Material-1
*Elastic
210000., 0.3
*Step, name=Step-1
*Static
1., 1., 1e-05, 1.
*Boundary
Nodeset1, 1, 2
Nodeset2, 1, 1
*Dsload
Surface1, P, -9.5
*Restart, write, frequency=0
*Output, field, variable=PRESELECT
*Output, history, variable=PRESELECT
*End Step
```

To run Abaqus *on a command line, first you have to get a* `DOS` *shell, as follows:* `On Windows 7` *click* `Start`*, then on the* `Search Programs and Files` *window, type* `cmd`*, then hit Enter. Then, on the* `DOS` *shell, do this:*

```
    >cd c:\SIMULIA
    >mkdir User
    >cd User
```

where `cd` *stands for change directory, and* `mkd` *creates a new folder (mkd stands for 'make directory').*[1]

Next, copy the file `Ex_2.2.inp` *from [5] to the folder* ***c: \Simulia\User****, or type the commands listed above using Notepad or your favorite text editor.*[2] *The command line to run* Abaqus *is:*

```
    >abaqus job=job1 input=Ex_2.2
```

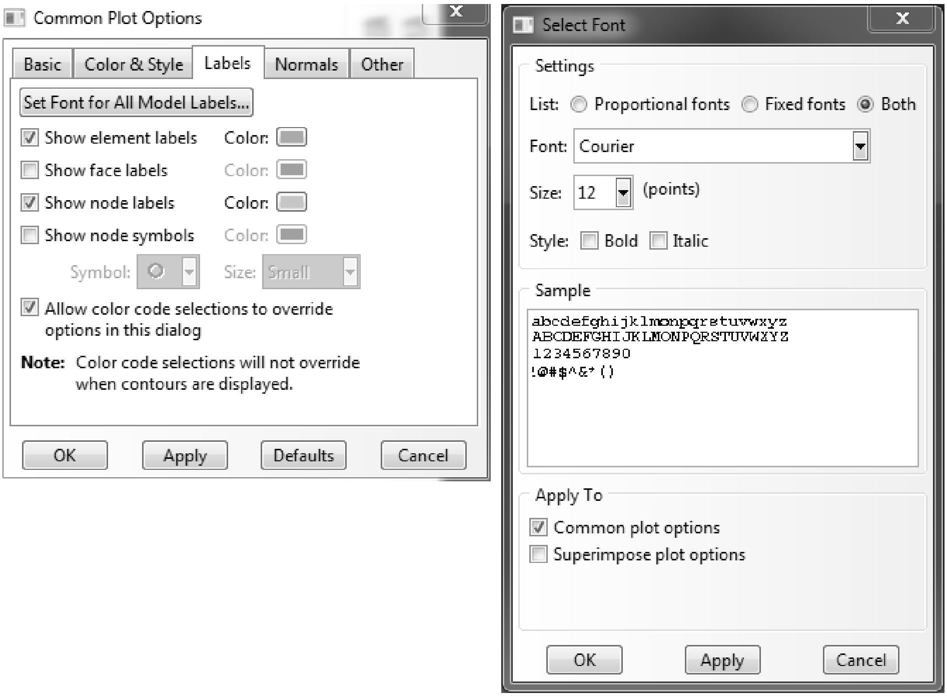
Now to start `CAE` *on* `Windows 7`*, click* `Start`*, then* `All Programs`*, then* `Abaqus 6.xx`*, then* `Abaqus/CAE`*. Once CAE starts, close the pop-up window. Then, on the Menu bar at the top of the screen*[3] *click* `File`*,* `Open`*, navigate to the* ***c: \Simulia\User\*** *folder, change the File Filter to* `*.odb`*, and select* `Job1.odb`*. At this point you should see the mesh, as shown in Fig. 2.7.*

To display element and node numbers do this: On the top Menu, `Options`*,* `Common`*,* `Labels` *(see Fig. 2.8), checkmark* `Show element labels` *and* `Show node labels`*. Also in the same window, to increase the font of the labels do this:* `Set Font`*. For future reference, the instructions given in this paragraph are summarized below (see notation in Table 2.1).*

[1] If your system administrator has restricted write access to `c:\SIMULIA` then make your user directory at some other unrestricted location such as `c:\TEMP` or see Appendix C for additional details.

[2] We recommend Notepad++ available free at `http://notepad-plus-plus.org/`.

[3] Also called display, monitor, or window.



Common Plot Options

☑ Show element labels Color:
□ Show face labels Color:
☑ Show node labels Color:
□ Show node symbols Color:
Symbol: Size: Small
☑ Allow color code selections to override options in this dialog

Note: Color code selections will not override when contours are displayed.

Select Font

Settings

List: ○ Proportional fonts ○ Fixed fonts ● Both

Font: Courier

Size: 12 (points)

Style: □ Bold □ Italic

Sample

```
abcdefghijklmnopqrstuvwxyz
ABCDEFGHIJKLMNOPQRSTUVWXYZ
1234567890
!@#$^&*()
```

Apply To

☑ Common plot options
□ Superimpose plot options

Fig. 2.8: To display element and node numbers in a larger font size.

```
Menu: File,
    Open [c:\Simulia\User\], File Filter [*.odb], File Name [Job1.odb]
Menu: Options,
    Common, Labels,
        # checkmark [Show element labels]
        # checkmark [Show node labels]
        Set Font for All Model Labels, Size [24], OK
    OK
```

Solid Modeling

With solid modeling, the user creates a geometric representation of the geometry using solid model constructs, such as volumes, areas, lines, and points. Boundary conditions, loads, and material properties can be assigned to parts of the solid model before meshing. In this way, re-meshing can be done without losing, or having to remove, the loads and boundary conditions. The models are meshed just prior to the solution. Solid modeling is used in Example 2.3, p. 54.

Abaqus/CAE has all the capabilities to construct the solid model, mesh, solve the problem, and visualize the results within the same CAE environment. More advanced solid modeling software, such as SolidWorks [11], can be used to generate the solid model, then import into CAE for finalizing the boundary conditions and loads, invoking the solver, and finally visualizing the results. A brief description of

Table 2.1: Pseudocode notation used to list the acronyms and commands[a] used to describe the user interaction with *Abaqus*/CAE.

=	equal
==	identical
(+)	Expand item on left menu tree
(−)	Collapse item on left menu tree
[]	user input such as file name, value, etc.
OK	name of button used often to complete a series of commands
X	name of button used often to cancel a series of commands
Cont	Continue, is the name of a button which means OK
Dis	Dismiss, is the name of a button that means Cancel
Enter	is the Enter key on the keyboard
#	comment line
# pick	pick object as requested in the window at the bottom of the WS
>	DOS prompt
>>>	Python prompt
Esc	Escape key
ctrl-z	undo
ctrl-y	redo
F6	autofit to workspace
crtl-click	unselect from selection
shift-click	add to selection
mouse wheel	zoom in/out
CS	coordinate system
BC	boundary conditions
WD	work directory
WS	workspace

[a] *Commands* are actions, instructions, steps, mouse clicks, selections, input data, and all other operations performed by the user on the Graphical User Interface.

the CAE graphical user interface (GUI) is described next. The notation used to describe the steps necessary to construct a model in CAE are listed in Table 2.1.

The CAE Window

The CAE window[4] (see Fig. 2.7) is divided in regions as follows:

- Menu: The top menu bar (horizontally across the top of the window) is dynamic, i.e., it changes depending on what we are doing. Specifically, it changes according to the selection in the drop-down list Module, which is explained below.
- Toolbar: Under the menu bar, we have dynamic toolbars. You can un-dock them, one at at a time, to learn their names. They are dynamic because they change depending on what Module is selected.
- Trees: On the left you see two tabs, Model and Results; one for each of the two database trees. All the items in the trees can be right-clicked to access actions to modify the information stored in each item. The trees correspond to the database files .mdb and .odb, as follows:
  - Model: .mdb
  - Results: .odb
- Drop-down menus: To the right of Model and Results tabs, we have several dynamic drop-down menus, such as Module, Model, and Part.[5] `Module:` contains a number of selections, as follows:
  - Module:
    - Part
    - Property
    - Assembly
    - Step
    - Interaction
    - Load
    - Mesh
    - Job
    - Visualization
    - Sketch
- Workspace: The workspace covers the most of the screen. It is located under the drop-down menus, to the right of the Model/Results tabs.

[4] At this time, the user should have *Abaqus*/CAE open on a computer.

[5] Since the drop-down menus are dynamic, only Module: and ODB: are shown in Fig. 2.7.

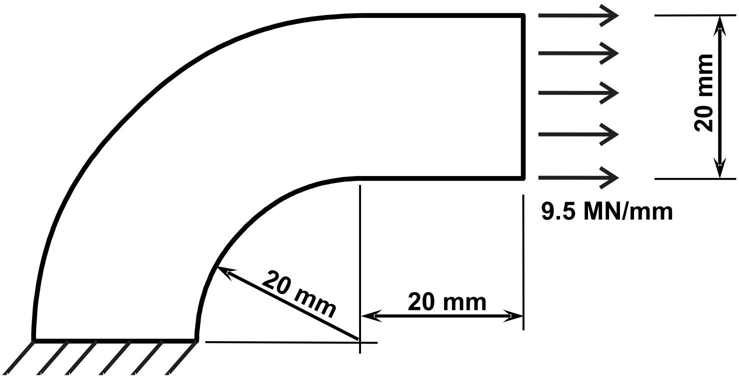


Fig. 2.9: Curved beam in Ex. 2.3.

- On a column between the Model/Result trees and the Workspace we have a series of icons, dynamically linked to the drop-down menus above. Icons have names that show up if you hover the cursor on them for a while. The icons are shortcuts to commands on the menu bar.

- Below the workspace there is an information window that toggles to a Python shell[6] by clicking on the tabs on the left (not shown in Fig. 2.7). The Python Shell is recognized by the prompt `>>>`

Example 2.3 *Generate the geometry shown in Fig. 2.9 using solid modeling techniques in* Abaqus*/CAE. Add boundary conditions and loads. Solve the problem and visualize the results. The thickness of the part is* 4.0 *mm. The material properties are* $E = 195,000\ MPa$, $\nu = 0.3$.

Solution to Example 2.3 *The solution to this example (Ex. 2.3) is* ***distributed among the various subsections*** *that explain the various steps involved in the modeling, solution, and visualization (see text in* ***italics*** *within pp. 54 to 75). Although it is possible to follow the example with the aid of the figures, it is best to have* Abaqus*/CAE running on a computer and to execute all the instructions while reading this section.*

To start Abaqus*/CAE on Windows 7, click* `Start`*,* `All Programs`*,* `Abaqus 6.xx`*,* `Abaqus/CAE`*.*[7] *Once CAE starts, close the pop-up window. Your screen should look more or less like Fig. 2.7 without the mesh.*

The first thing you want to do is to make sure the files for this example are saved to a separate folder, so that they don't get mixed up with other examples. The easiest way to do that is to save this example right now, even though it is empty,

[6] A *shell* is a process window that accepts literal (typed) commands. A Python shell accepts Python code. A DOS shell accepts DOS script, such as `cd`, and so on.

[7] Productivity tip: Pin the shortcut for Abaqus/CAE to the Windows Taskbar. To do this, drag the shortcut on top of the Windows Start button.

to a new folder, as follows. On the Menu bar at the top of the screen click File, Save As, *navigate to a new folder of your choice, or create one if you need to; then in the field* File name *enter* Ex_2.3, OK. *These steps are summarized below.*

```
Menu:   File, Save as, New directory [name of new directory],
        Select new directory, [name of file], OK
```

The first modeling task is to create one or more Parts, which represent the geometry of the model. Since this example is very simple, a single part will suffice, but more complex models will require several parts. There are many ways to define the geometry of a part. In this first example we use Points connected by Arcs and Lines. To work with parts, the Module has to be set to Part. Remember that

- ***Module:*** *refers to the drop-down box immediately left of the Model/Result tabs (see Fig. 2.7), and*
- ***Menu:*** *refers to the top menu bar running across the top of the CAE window.*

In this way, the selections described below will result in opening the dialog box shown in Fig. 2.10:

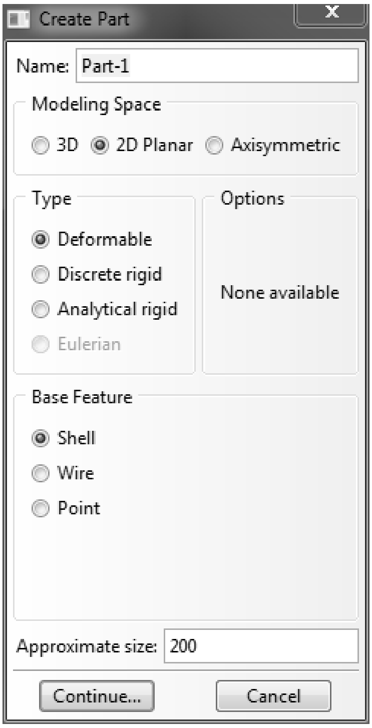
```
Module: Part
    Menu: Part, Create,
    Name[Part-1], 2D, Deformable, Shell, Approx size [200], Cont
    # sketch mode is now active
```

Next, we define Points and join them with Arcs, as summarized below. See Fig. 2.11 in conjunction with the step-by-step instructions below.

```
Menu: Add, Point,
    [0,0] # type in the input window below the WS, then [Enter]
    [-20,0]
    [-40,0]
    [0,20]
    [0,40]
    [20,20]
    [20,40]
    F6    # zoom to fit
Menu: Add, Arc, Center/Endpoints,
    # pick center point (see Fig. 2.11)
    # pick point left of center point
    # pick point above of center point, the arc is done
    # again pick center point (see Fig. 2.11)
    # pick point left-most of center point
    # pick point above-most of center point, the outer arc is done
    X # to finalize the Add feature (do not click Done)
```

At this point, the workspace should look like Fig. 2.11.

Sketch mode *is similar to SolidWorks [11]. Arcs, points, and other geometrical constructs are called* entities. *When adding/editing entities, you must follow the*



Name: Part-1

Modeling Space

○ 3D ● 2D Planar ○ Axisymmetric

Type

● Deformable
○ Discrete rigid
○ Analytical rigid
○ Eulerian

Options

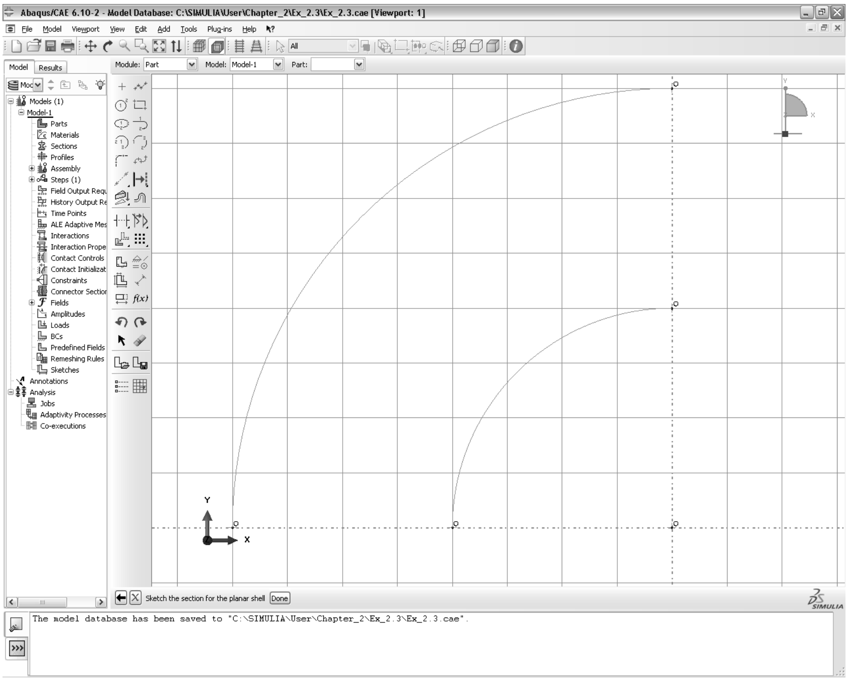
None available

Base Feature

● Shell
○ Wire
○ Point

Approximate size: 200

Fig. 2.10: The Create Part dialog box.



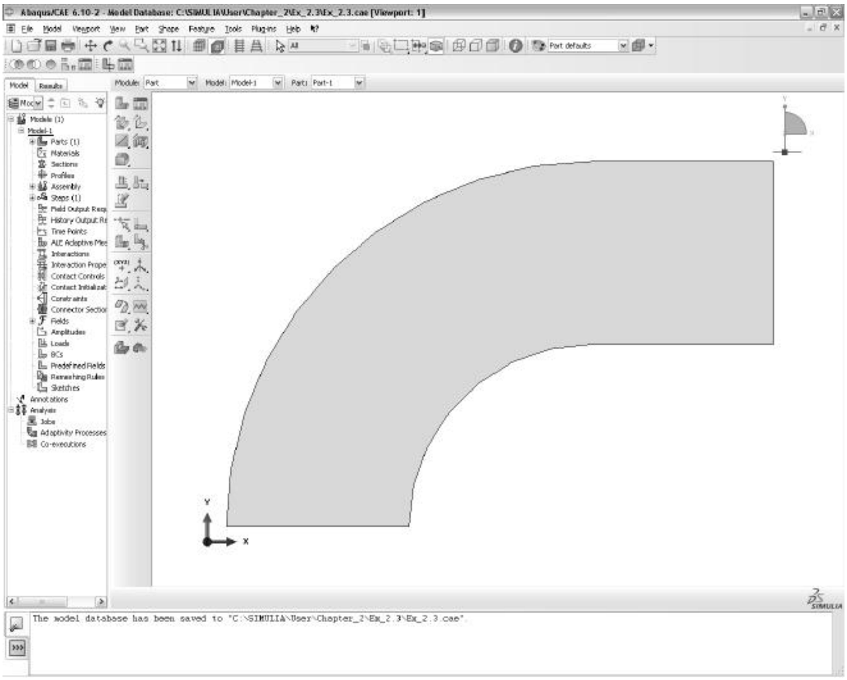
Abaqus/CAE 6.10-2 - Model Database: C:\SIMULIA\User\Chapter_2\Ex_2.3\Ex_2.3.cae [Viewport: 1]

Module: Part Model: Model-1 Part:

Sketch the section for the planar shell

```
The model database has been saved to "C:\SIMULIA\User\Chapter_2\Ex_2.3\Ex_2.3.cae".
```

Fig. 2.11: The two arcs are defined.



Abaqus/CAE 6.10-2 - Model Database: C:\SIMULIA\User\Chapter_2\Ex_2.3\Ex_2.3.cae [Viewport: 1]

Module: Part Model: Model-1 Part: Part-1

```
The model database has been saved to "C:\SIMULIA\User\Chapter_2\Ex_2.3\Ex_2.3.cae".
```

Fig. 2.12: The Part is finalized.

instructions displayed immediately below the WS. The ***X*** *situated to the left of the instructions line allows you to finalize the current operation, i.e., the* Add *feature in the set of instructions above.*

Still referring to Fig. 2.11, add lines to close the boundary of the figure, and once the figure is closed, the part is created:

```
Menu: Add, Lines, Connected,
    # pick two nodes to close bottom left of the part
    X   # to finalize the Line drawing
Menu: Add, Lines, Connected,
    # pick 4 nodes in sequence to close rectangle at top right
    X   # to finalize the Line drawing
Done    # to finish the sketch, thus creating the part
```

At this point, the workspace should look like Fig. 2.12.

2.3.2 Material and Section Properties

Parts must be associated to materials. Depending on the analysis, material properties can be linear (linear elastic analysis) or nonlinear (e.g. damage mechanics analysis), isotropic or orthotropic, constant or temperature-dependent. Entering the correct materials properties is one of the most important aspects of a successful analysis of composite materials. A great deal of attention is devoted to material properties in the rest of the textbook. For now it will suffice to illustrate the process using a linear elastic, isotropic material. For structural analysis, elastic properties must be defined according to Section (1.12). Other mechanical properties, such as strength, density, and thermal expansion coefficients are optional and their definition depends on the objectives of the analysis.

In Abaqus/CAE, materials are entered in `Module: Property`, `Menu: Material`. *The instructions below will bring out a pop-up window as shown in Fig. 2.13, where the materials properties are entered, as follows:*

```
Module: Property
    Menu: Material, Create,
        Name [Material-1], Mechanical, Elasticity, Elastic,
        Type, Isotropic [195000, 0.3], OK
```

All elements need material properties, but *structural* elements need additional parameters that vary with the type of element. These parameters result from analytical integration of the 3D governing equations while formulating the element. For example, the cross-section area A appears in (2.1) because the 3D partial differential equations have been integrated over the cross section of the rod to arrive at the ordinary differential equation (2.1).

Beam elements require cross-section area and moments of inertia. *Laminated shell* elements require the laminate stacking sequence (LSS, see Figure 3.4, p. 96). *Continuum* elements such as 3D solid elements (see Table 2.2) do not require additional parameters, only material properties, because the geometry is fully described by the mesh. However, continuum elements representing laminated composites still require the LSS.

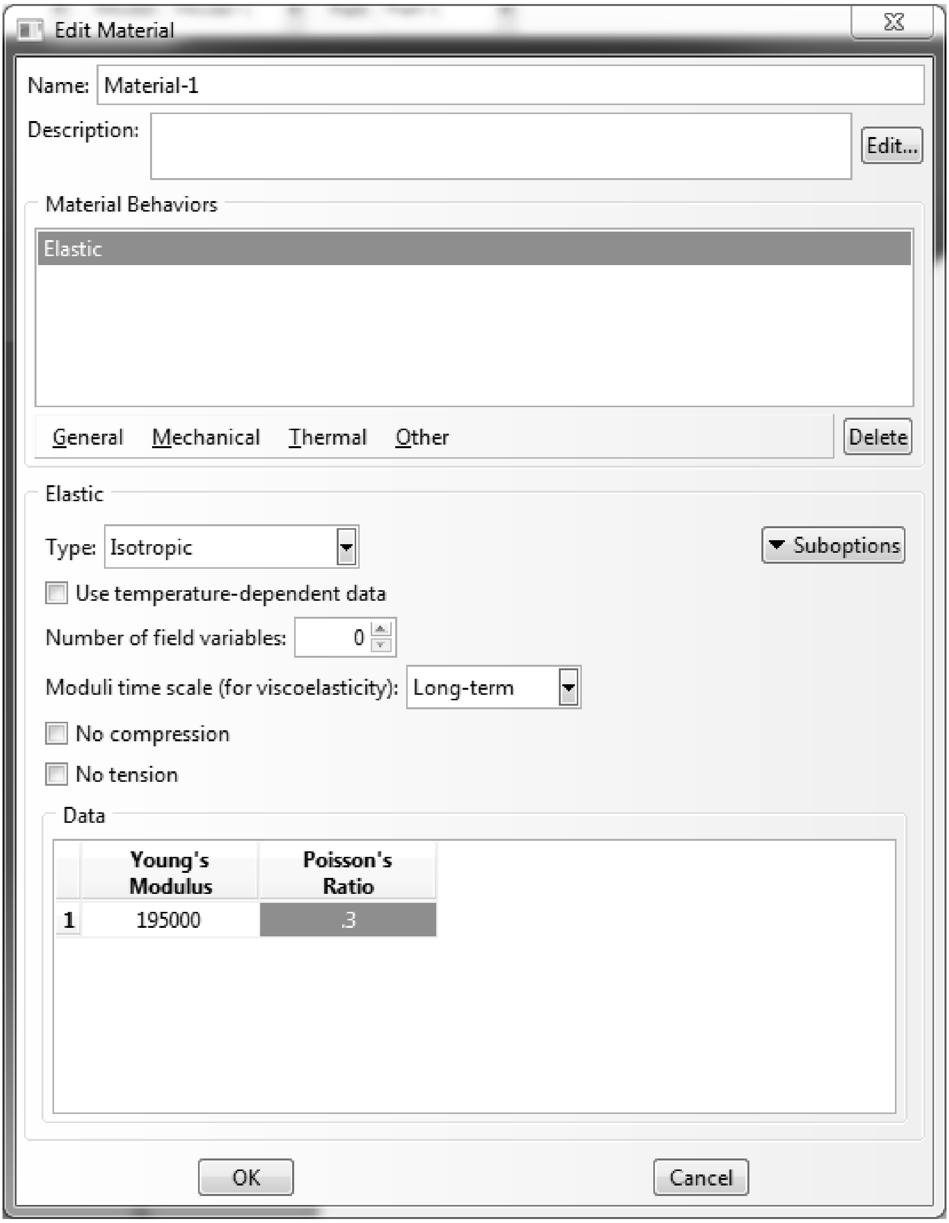
Conventional shells and beams are typical *structural* elements. The word *conventional* is used also by *Abaqus* to emphasize shell elements that are not continuum. On the other hand, continuum shells are continuum elements with kinematic constrains introduced in order to represent shell behavior.

Therefore, besides a `Material`, one needs to create a `Section` to provide the additional parameters required by structural elements. In *Abaqus*, the additional parameters are called *section parameters*; in *ANSYS* they are called *section constants*.[8] Continuum elements do not require section parameters, but a `Section` has to be created anyway.

In this simple example, section parameters are used to define the state of plane stress for the part shown in Fig. 2.9. A complex model may have several different sections. Sections are created in the same Module: Property, as follows:

```
Menu: Section, Create,
    # plane stress/strain is considered solid even though
    # pstres needs a thickness
    # pstran does not
    # shells are not solids, they need a thickness
    # beams are not solids, they need area and moments of inertia
    Name [Section-1], Solid, Homogeneous, Cont
```

[8] Section constants were called “real constants” in older versions of *ANSYS*.



Edit Material

Name: Material-1

Description:

Material Behaviors

Elastic

Elastic

Type: Isotropic

☐ Use temperature-dependent data

Number of field variables: 0

Moduli time scale (for viscoelasticity): Long-term

☐ No compression

☐ No tension

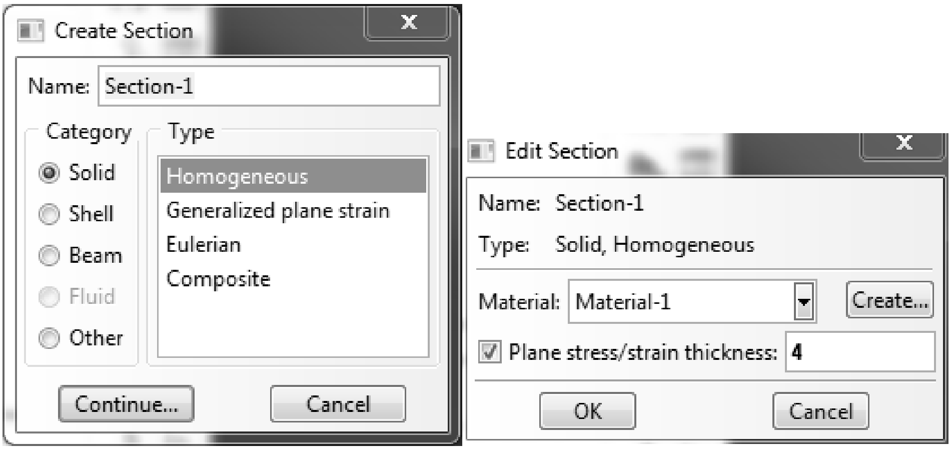
Data

	Young's Modulus	Poisson's Ratio
1	195000	.3

Fig. 2.13: Material properties window.

Table 2.2: Some of the elements available in *Abaqus* and *ANSYS*.

Abaqus	**ANSYS**	**Nodes**	**DOF**	**Element Description**
Structural elements				
T2D2	LINK180	2	u_X u_Y	line bar/truss, 2D space
T3D2	LINK180	2	u_X u_Y u_Z	line bar/truss, 3D space
-	COMBIN14	2	u_X u_Y u_Z	spring/damper, 3D space
B21	BEAM188	2	u_X u_Y θ_X θ_Y	line beam in 2D space
B31	BEAM188	2	u_X u_Y u_Z θ_X θ_Y θ_Z	line beam in 3D space
CPE4R	PLANE182	4	u_X u_Y	solid quadrilateral in 2D space
CPE8R	PLANE183	8	u_X u_Y	solid quadrilateral in 2D space
S4R	SHELL181	4	u_X u_Y u_Z θ_X θ_Y θ_Z	shell quadrilateral in 3D space (conventional)
S8R	SHELL281	8	u_X u_Y u_Z θ_X θ_Y θ_Z	shell quadrilateral in 3D space (conventional)
S8R5	-	4	u_X u_Y u_Z θ_X θ_Y	thin shell quadrilateral in 3D space (conventional)
Continuum elements				
C3D8	SOLID185	8	u_X u_Y u_Z	solid hexahedra in 3D space
C3D20	SOLID186	20	u_X u_Y u_Z	solid hexahedra in 3D space
SC8R	SOLSH190	8	u_X u_Y u_Z	shell hexahedra in 3D space (continuum)



Create Section

Name: Section-1

Category: ● Solid ○ Shell ○ Beam ○ Fluid ○ Other

Type: Homogeneous (selected), Generalized plane strain, Eulerian, Composite

Edit Section

Name: Section-1

Type: Solid, Homogeneous

Material: Material-1

☑ Plane stress/strain thickness: 4

Fig. 2.14: The Create Section and Edit Section windows define the Section.

```
    Material: Material-1,
    Plane stress/strain thickness [4.0], OK
```

The Create Section and Edit Section windows, shown in Fig. 2.14, appear in sequence. They are used to define the section.

Finally, we need to tell the software what are the section properties for each part. This is done by assigning *Sections to Parts. In this simple model there is only one part and only one section, so the assignment is trivial, but it has to be done anyway. Note that the assignment is done in the same module, i.e., the Property module. The procedure is illustrated in Fig. 2.15 and summarized as follows:*

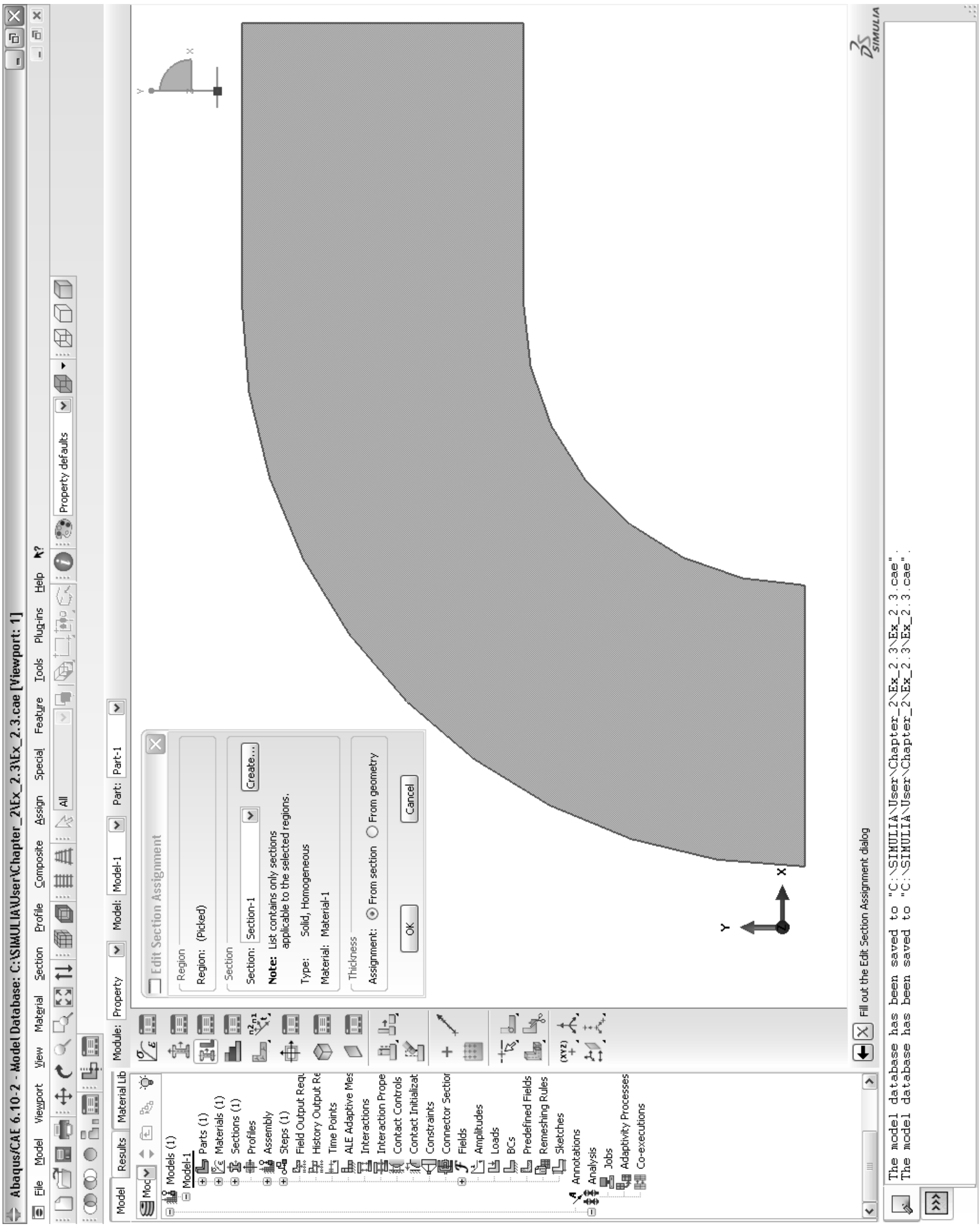
```
Menu: Assign, Section
    # pick all (click on part), Done
    OK
```

2.3.3 Assembly

If more than one part exists, assembly is necessary to put the parts together into what is called an *assembly*, which represents the physical object you are trying to analyze.

During assembly it is possible to specify how the mesh is related to the parts. That is, the mesh can be dependent *or* independent *of the part. A dependent mesh is tied to the part. So, if the part is used (i.e., instanced) multiple times in an assembly, all the instances of the part will be meshed identically. And independent mesh means that each instance of the part will have to be meshed independently. The latter provides flexibility to refine the mesh for some instances of the part at the expense of more work when it comes the time to mesh. Since this example has only one part, the assembly process is trivial, as follows (see also Fig. 2.16):*

```
Module: Assembly
Menu: Instance, Create,
    Independent, OK
```



Abaqus/CAE 6.10-2 - Model Database: C:\SIMULIA\User\Chapter_2\Ex_2.3\Ex_2.3.cae [Viewport: 1]

Module: Property Model: Model-1 Part: Part-1

Edit Section Assignment

Region

Region: (Picked)

Section

Section: Section-1

Note: List contains only sections applicable to the selected regions.

Type: Solid, Homogeneous

Material: Material-1

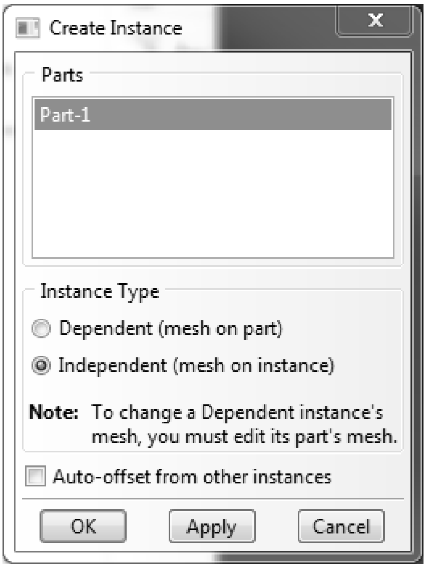
Thickness

Assignment: ● From section ○ From geometry

Fill out the Edit Section Assignment dialog

```
The model database has been saved to "C:\SIMULIA\User\Chapter_2\Ex_2.3\Ex_2.3.cae".
The model database has been saved to "C:\SIMULIA\User\Chapter_2\Ex_2.3\Ex_2.3.cae".
```

Fig. 2.15: Assigning a section to the part.



Create Instance

Parts

Part-1

Instance Type

○ Dependent (mesh on part)

● Independent (mesh on instance)

Note: To change a Dependent instance's mesh, you must edit its part's mesh.

□ Auto-offset from other instances

Fig. 2.16: Dialog box to create an instance of a part.

2.3.4 Solution Steps

Next, the analysis process is normally broken down into several *steps*, each representing different loading and constraint conditions. The minimum number of steps is two: an initial step and at least one additional step. No loads can be applied on the initial step, only boundary conditions.

The current example uses just those two steps, as follows:

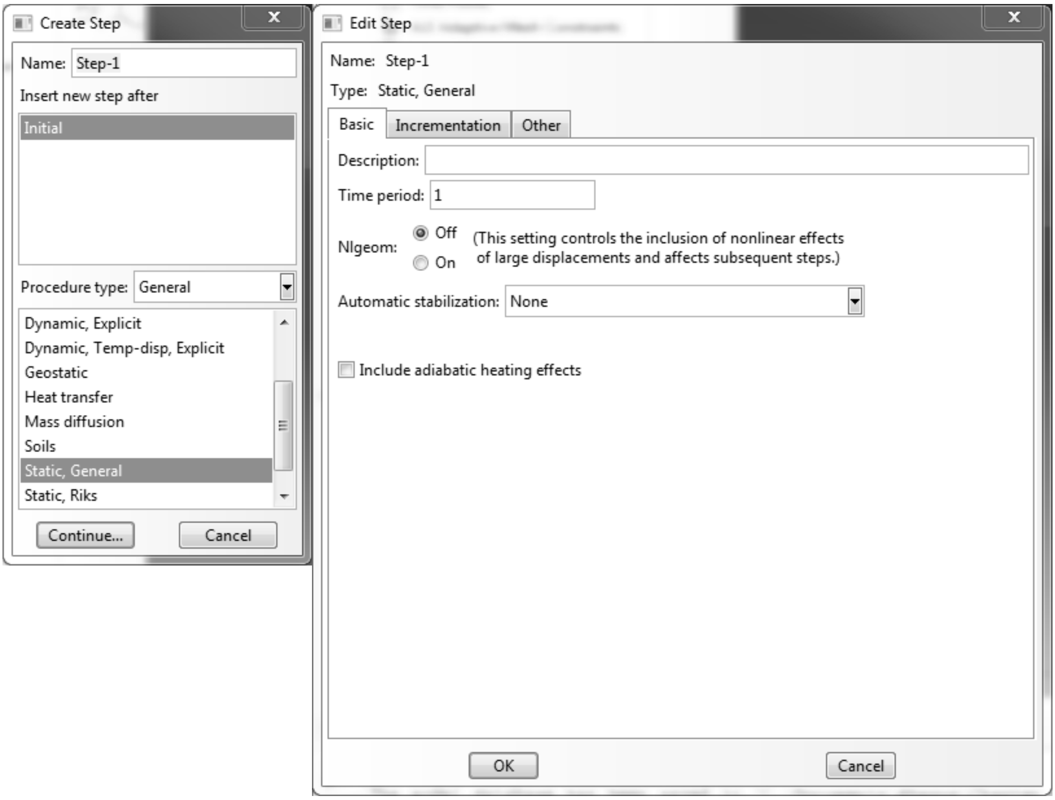
```
Module: Step
Menu: Step, Create,
    Name [Step-1], General, Static/General, Cont
    OK
```

The dialog box for Create Step is followed by one for Edit Step to complete the data entry for this step, as shown in Fig. 2.17.

2.3.5 Loads

In structural analysis, loads are defined by forces, pressures, inertial forces (as gravity), and specified displacements, all applied to the model. Specification of different kinds of loads for the FE model is explained in the following sections. The reactions obtained by fixing a nodal degree of freedom (displacements and rotations) are discussed also.

Loads can be applied on nodes by means of concentrated forces and moments, as shown in Example 2.4, p. 75. Also, loads can be distributed over the elements as: surface loads, body loads, inertia loads, or other coupled-field loads (for example, thermal strains). Surface loads are used in Example 2.5, p. 77.



Create Step

Name: Step-1

Insert new step after

Initial

Procedure type: General

Static, General

Edit Step

Name: Step-1

Type: Static, General

Description:

Time period: 1

Nlgeom: ● Off ○ On (This setting controls the inclusion of nonlinear effects of large displacements and affects subsequent steps.)

Automatic stabilization: None

□ Include adiabatic heating effects

Fig. 2.17: Dialog box to create a Step.

A surface load is a distributed load applied over a surface, for example a pressure. A body load is a volumetric load, for example expansion of material by temperature increase in structural analysis. Inertia loads are those attributable to the inertia of a body, such as gravitational acceleration, angular velocity, and acceleration.

A concentrated load applied on a node is directly added to the force vector. However, the element interpolation functions are used to compute the equivalent forces vector due to distributed loads.

In Abaqus*/CAE, loads and boundary (support) conditions for the structure are specified with the Loads module, as follows:*

```
Module: Load
Menu: Load, Create,
    Name [load-1], Mechanical, Pressure, Cont
    # pick the vertical edge on the right, Done
    Magnitude [-9.5], OK
```

The dialog box for Create Load (top left in Fig. 2.18) is followed by the selection of the surfaces over which the load (or bc) are to be applied (note dialog box below the x-y-z triad in Fig. 2.18), then followed by the Edit Load box to enter the values for the load (top right in Fig. 2.18).

2.3.6 Boundary Conditions

The boundary conditions are the known values of the degrees of freedom (DOF) on the boundary. In structural analysis, the DOF are displacements and rotations. With this information, the software knows which values of $\underline{a}$ in (2.46) are known or unknown.

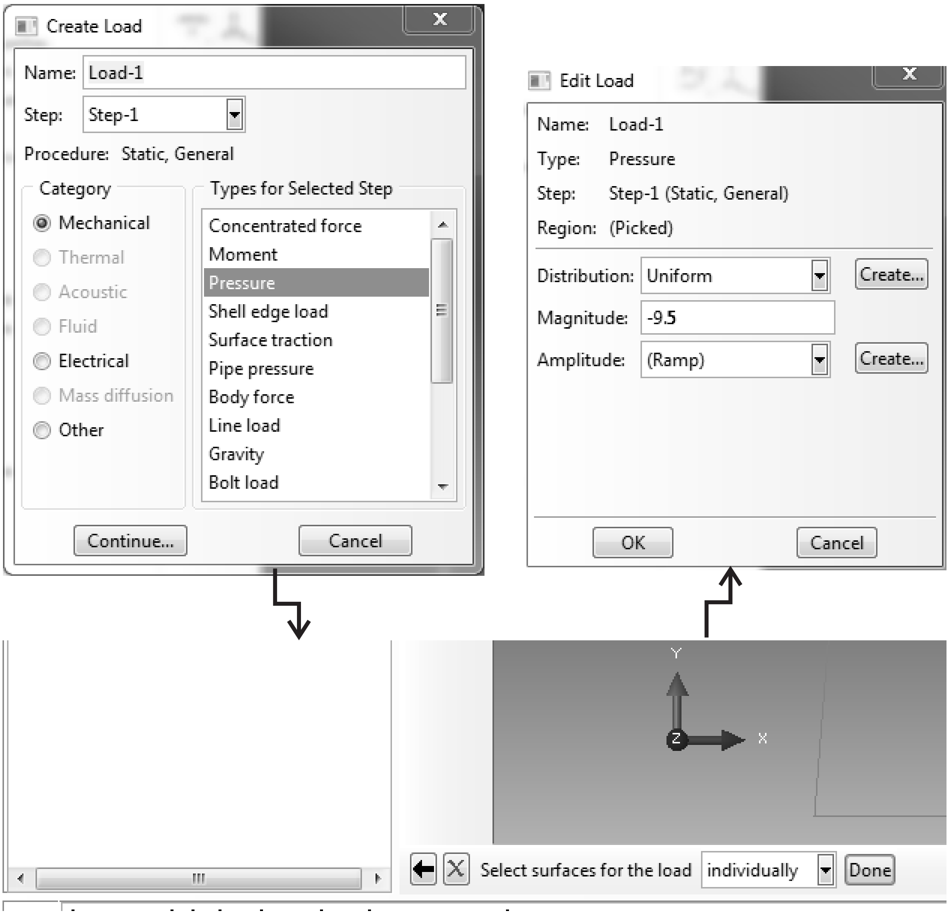
Constrained Displacements and Rotations

In general, a node can have more than one DOF. For example, if the FE model uses beam elements in 2D space, there are three DOF: the horizontal displacement, the vertical displacement, and the rotation around an axis perpendicular to the plane. Constraining different sets of DOF results in different boundary conditions being applied. In the 2D beam element case, constraining only the horizontal and vertical displacements results in a simple support, but constraining all the DOF results in a clamped condition.

Symmetry Conditions

Symmetry conditions can be applied to reduce the size of the model without loss of accuracy. Four types of symmetry must exist concurrently: symmetry of geometry, boundary conditions, material, and loads. Under these conditions the solution will also be symmetric. For example, symmetry with respect to the $y - z$ plane means that the nodes on the symmetry plane have the following constraints

$$u_x = 0 \quad ; \quad \theta_y = 0 \quad ; \quad \theta_z = 0 \tag{2.47}$$



Create Load

Name: Load-1

Step: Step-1

Procedure: Static, General

Category

● Mechanical
○ Thermal
○ Acoustic
○ Fluid
○ Electrical
○ Mass diffusion
○ Other

Types for Selected Step

Concentrated force
Moment
Pressure
Shell edge load
Surface traction
Pipe pressure
Body force
Line load
Gravity
Bolt load

Edit Load

Name: Load-1

Type: Pressure

Step: Step-1 (Static, General)

Region: (Picked)

Distribution: Uniform

Magnitude: -9.5

Amplitude: (Ramp)

Select surfaces for the load individually

Fig. 2.18: Dialog box to create a Load.

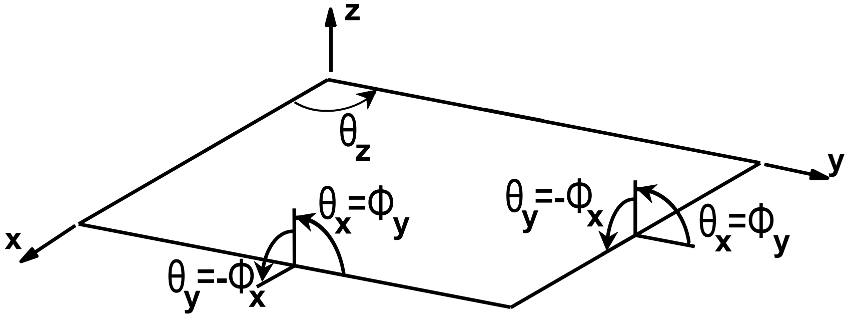


Fig. 2.19: Convention for rotations of a plate or shell.

where u_x is the displacement along the x-direction, θ_y and θ_z are the rotations around the $y-$ and $z-$axis, respectively (Figure 2.19). Note that the definition of rotations used in shell theory (ϕ_i, see Section 3.1) is different than the usual definition of rotations θ_i that follows the right-hand rule. All rotations in *Abaqus*[9] are described using right-hand-rule rotations θ_i. Symmetry boundary conditions on nodes in the symmetry plane involve the restriction of DOF translations out-of-plane with respect to the symmetry plane and restriction of the DOF rotations in-plane with respect to the symmetry plane. Symmetry boundary conditions are used in Example 2.5, p. 77.

Antisymmetry Conditions

Antisymmetry conditions are similar to the symmetry conditions. They can be applied when the model exhibits antisymmetry of loads but otherwise the model exhibits symmetry of geometry, symmetry of boundary conditions, and symmetry of material. Antisymmetry boundary conditions on nodes in the antisymmetry plane involve restriction of DOF translations in the antisymmetry plane and restriction of DOF rotations out-of-plane with respect to the antisymmetry plane.

Periodicity Conditions

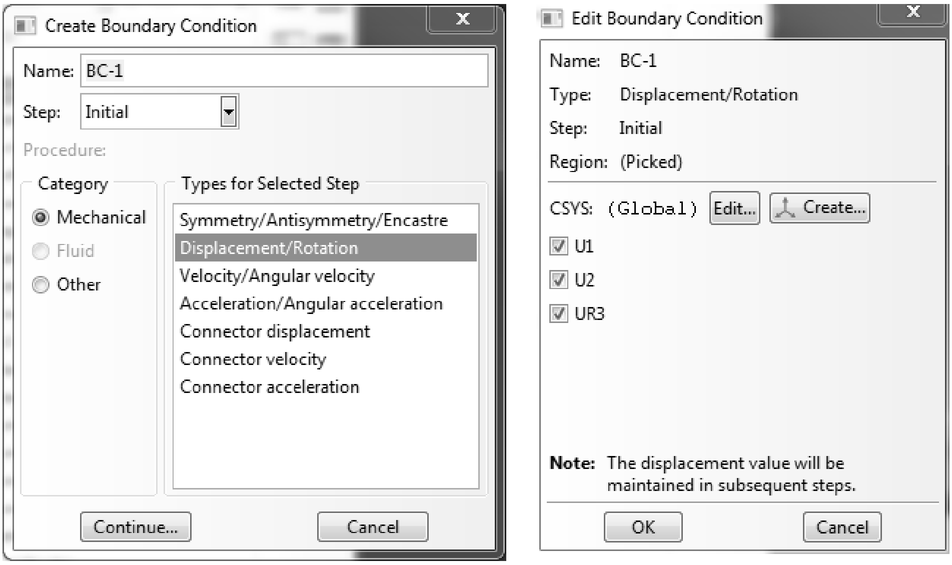
When the material, load, boundary conditions, and geometry are periodic with period $(x, y, z) = (2a_i, 2b_i, 2c_i)$, only a portion of the structure needs to be modeled, with dimensions $(2a_i, 2b_i, 2c_i)$. The fact that the structure repeats itself periodically means that the solution will also be periodic. Periodicity conditions can be imposed by different means. One possibility involves using constraint equations (CE) between DOF (see Section 6.2) or using Lagrange multipliers.

Boundary Conditions in *Abaqus*/CAE

In Abaqus*/CAE, the dialog box for Create a Boundary Condition is followed by the Edit Boundary Condition dialog, as shown in Fig. 2.20. The dialog box is reached as follows:*

```
Module: Load
Menu: BC, Create,
    Name [BC-1], Step: Initial, Mechanical, Displacement/Rot, Cont
    # pick the horizontal edge on bottom, Done
    # checkmark U1, U2, UR3, OK
    Step: Step-1  # to see the load and BC together
```

[9] Quote "Except for axisymmetric elements, the degrees of freedom are always referred to as follows: 1: x-displacement 2: y-displacement 3: z-displacement 4: Rotation about the x-axis, in radians 5: Rotation about the y-axis, in radians 6: Rotation about the z-axis, in radians ... Here the x-, y-, and z-directions coincide with the global X-, Y-, and Z-directions, respectively; however, if a local transformation is defined at a node (see Transformed coordinate systems [7, Section 2.1.5]), they coincide with the local directions defined by the transformation." End quote [7].



Create Boundary Condition

Name: BC-1

Step: Initial

Procedure:

Category

● Mechanical

○ Fluid

○ Other

Types for Selected Step

Symmetry/Antisymmetry/Encastre
Displacement/Rotation
Velocity/Angular velocity
Acceleration/Angular acceleration
Connector displacement
Connector velocity
Connector acceleration

Continue... Cancel

Edit Boundary Condition

Name: BC-1

Type: Displacement/Rotation

Step: Initial

Region: (Picked)

CSYS: (Global) Edit... Create...

☑ U1

☑ U2

☑ UR3

Note: The displacement value will be maintained in subsequent steps.

OK Cancel

Fig. 2.20: Dialog box to create Boundary Conditions.

To see the loads together with the BC, one must change the visualized step to Step-1, because on Step: Initial, only the BC are present. To change the visualized step, look for the `Step` *drop-down box to the right of the* `Module` *drop-down menu. Once the change is made, the WS should look like Fig. 2.21.*

2.3.7 Meshing and Element Type

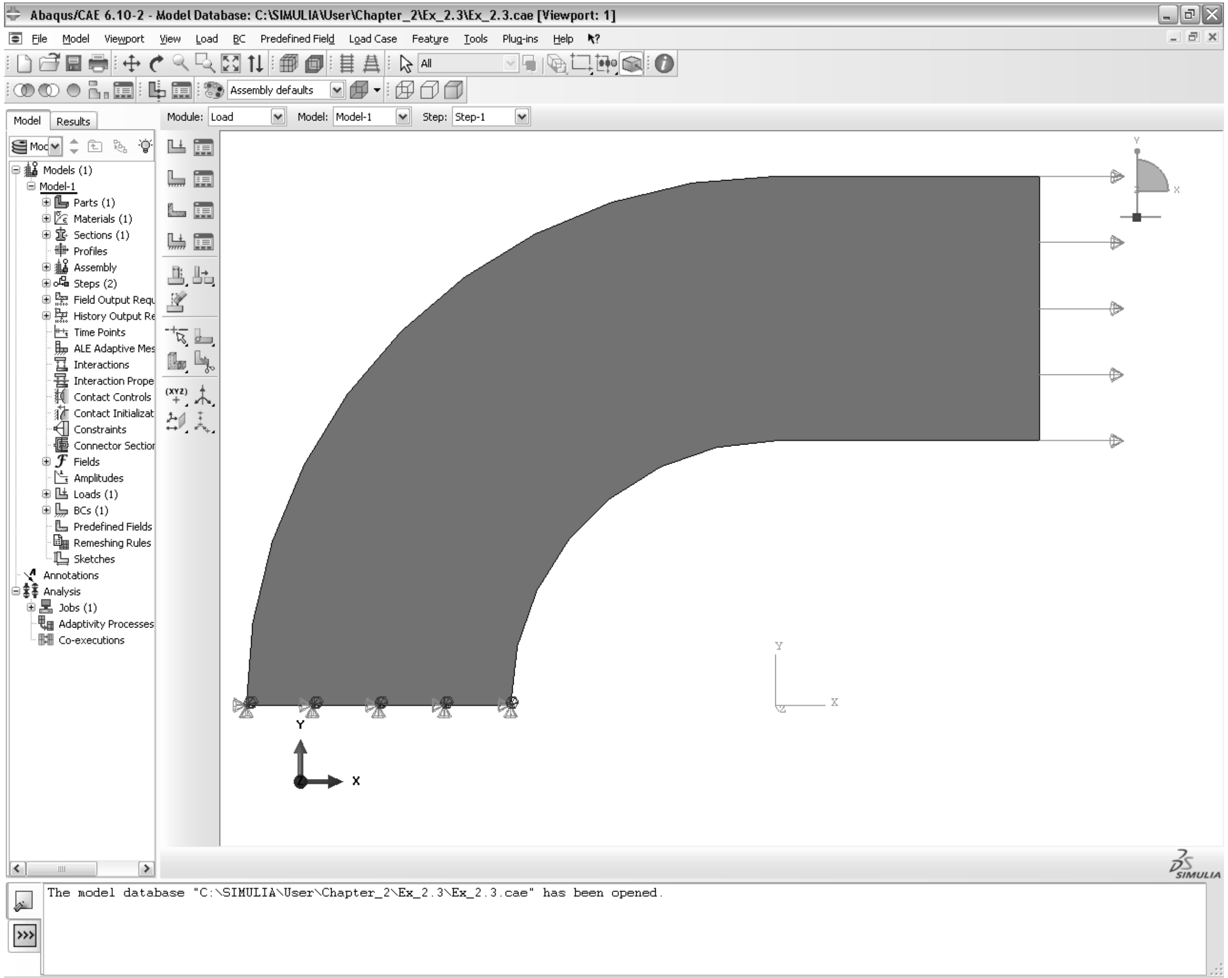
Next, the assembly needs to be meshed. There are many ways to mesh a model. For this example we use a very simple approach, called global seeding, as follows:

```
Module: Mesh
Menu:   Seed, Instance,
        Approx global size [5.0], Apply, OK
Menu:   Mesh, Controls, Quad, Structured, OK
```

Global seeding, as shown in the left dialog box in Fig. 2.22, is perhaps the easiest approach to specify mesh density. It is best to click Apply before OK. This provides an opportunity to see the seeded part, and thus to modify it, before committing to a given seed pattern. Of course everything can be edited, but that is for later.

Then, on the right dialog box in Fig. 2.22, Mesh Control is used to specify the element characteristics.

Finite Element Analysis (FEA) programs have an *element library* that contains many different element types. The element type determines the element formulation used. For example, the degree of freedom set, the interpolation functions, whether the element is for 2D or 3D space, etc. The element type identifies the element category: bar/rod tensile-compression, beam bending, solid, shell, laminate shell, etc. Each commercial code identifies element formulations with different labels.



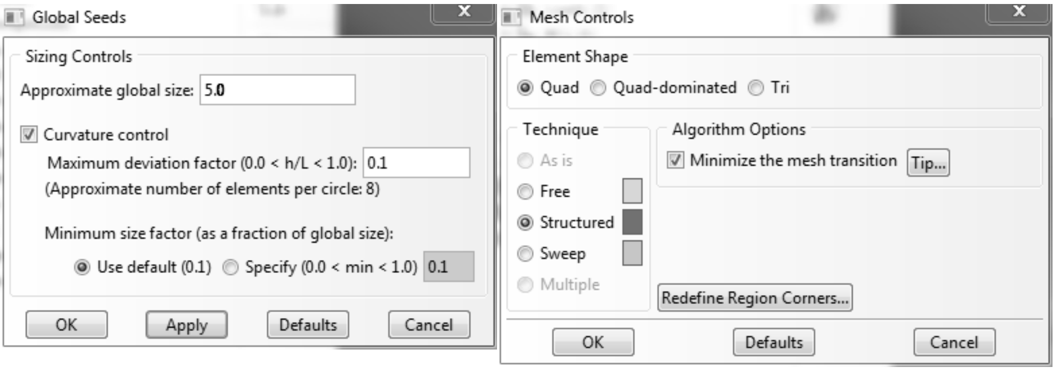
All

Assembly defaults

Module: Load Model: Model-1 Step: Step-1

```
The model database "C:\SIMULIA\User\Chapter_2\Ex_2.3\Ex_2.3.cae" has been opened.
```

Fig. 2.21: Displaying the loads and boundary conditions.



Global Seeds

Sizing Controls

Approximate global size: 5.0

☑ Curvature control

Maximum deviation factor (0.0 < h/L < 1.0): 0.1

(Approximate number of elements per circle: 8)

Minimum size factor (as a fraction of global size):

● Use default (0.1) ○ Specify (0.0 < min < 1.0) 0.1

Mesh Controls

Element Shape

● Quad ○ Quad-dominated ○ Tri

Technique

○ As is

○ Free

● Structured

○ Sweep

○ Multiple

Algorithm Options

☑ Minimize the mesh transition

Fig. 2.22: Using Global Seeds (left dialog box) and Mesh Control (right dialog box).

Identification labels and basic characteristics of a few element formulations are shown in Table 2.2. Also, each element type has different options. For example, on a planar solid element, an option allows one to choose between plane strain and plane stress analysis.

The element type is selected in the Element Type dialog box shown in Fig. 2.23. The last step is to finalize (instance) the mesh. The procedure is summarized as follows:

```
Menu: Mesh, Element type, Standard, Linear, Plane stress, OK
Menu: Mesh, Instance, Yes
```

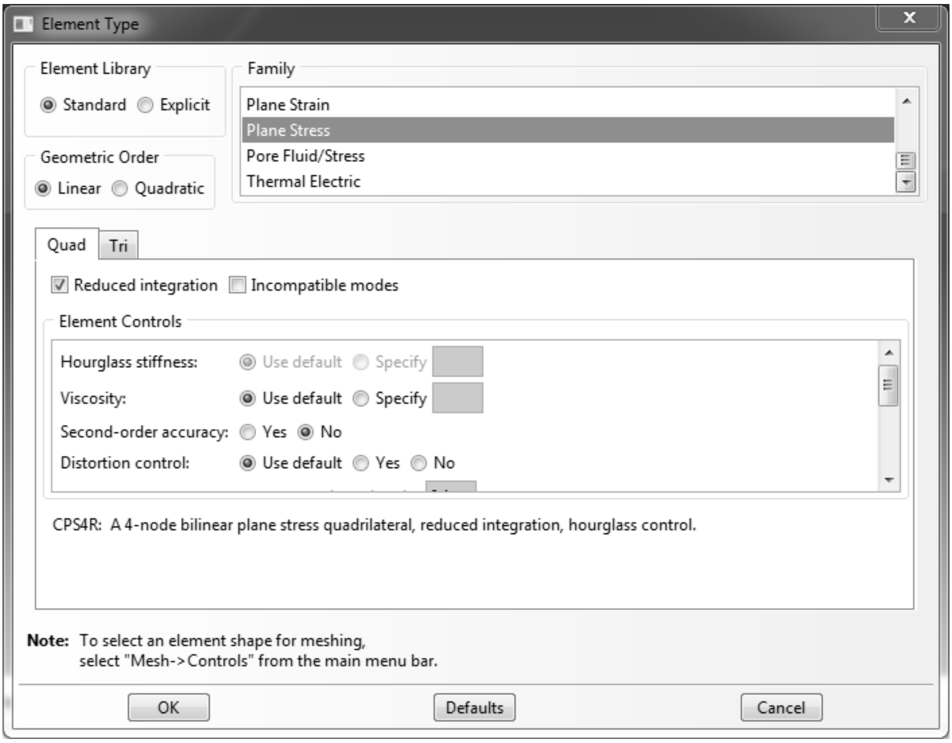
Finally, your mesh should look like the one displayed in Fig. 2.24.

2.3.8 Solution Phase

In the solution phase of the analysis, the solver subroutine included in the finite element program solves the simultaneous set of equations (2.46) that the finite element method generates. Usually, the primary solution is obtained by solving for the nodal degree of freedom values, i.e., displacements and rotations. Then, derived results, such as stresses and strains, are calculated at the integration points. Primary results are called *nodal solutions* and derived results are called *element solutions.*

Several methods of solving the system of simultaneous equations are available. Some methods are better for larger models, others are faster for nonlinear analysis, others allow one to distribute the solution by parallel computation. Commercial finite element programs solve these equations in batch mode. The frontal direct solution method is commonly used because it is rather efficient for FEA. When the analysis is nonlinear, the equations must be solved repeatedly, thus increasing the computational time significantly.

To solve the model, one must create a Job. The job is then submitted from within CAE for execution by the Abaqus *processor. The CAE submit process is simply a*



Element Type

Element Library
● Standard ○ Explicit

Family
Plane Strain
Plane Stress
Pore Fluid/Stress
Thermal Electric

Geometric Order
● Linear ○ Quadratic

Quad | Tri

☑ Reduced integration ☐ Incompatible modes

Element Controls
Hourglass stiffness: ● Use default ○ Specify
Viscosity: ● Use default ○ Specify
Second-order accuracy: ○ Yes ● No
Distortion control: ● Use default ○ Yes ○ No

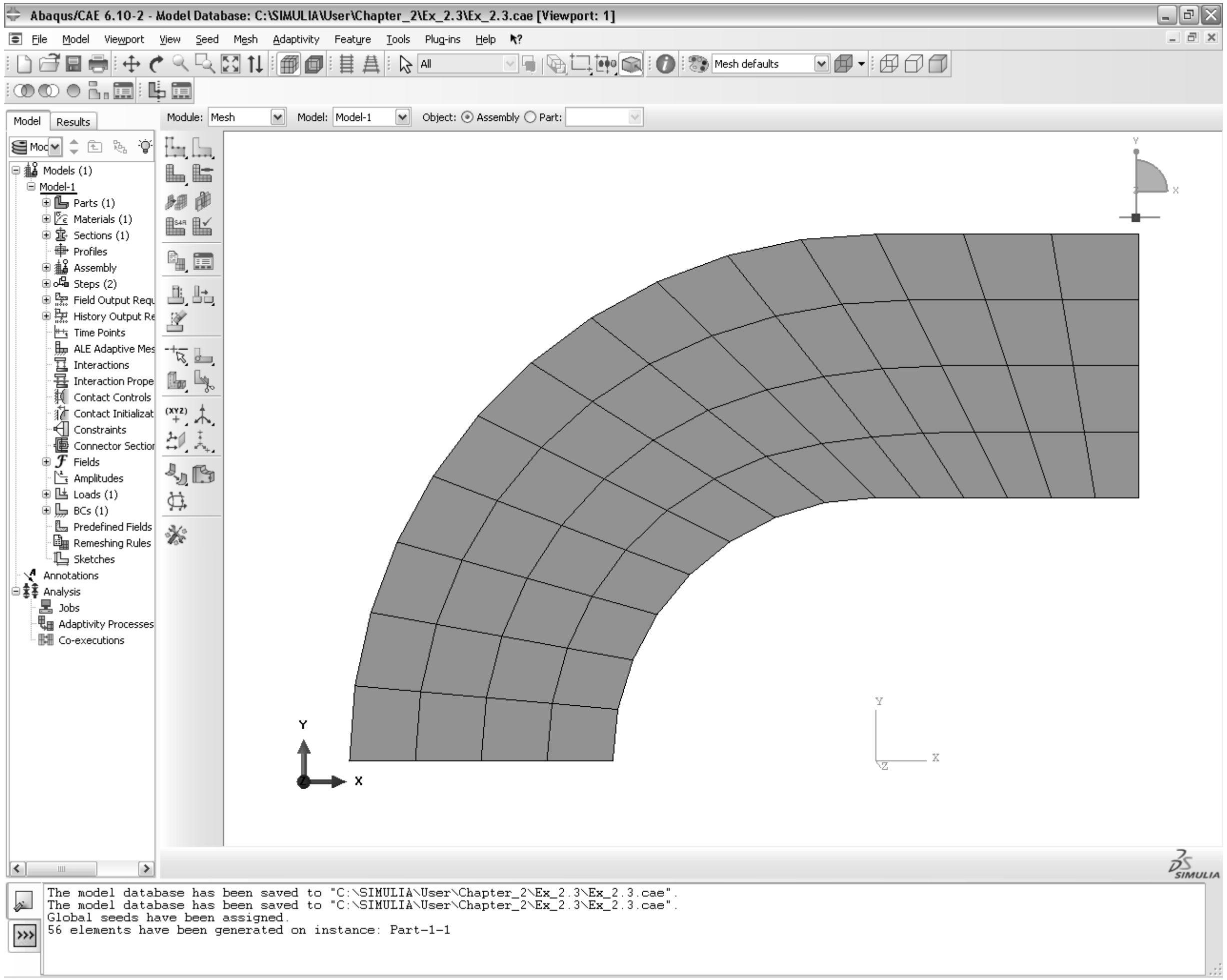
CPS4R: A 4-node bilinear plane stress quadrilateral, reduced integration, hourglass control.

Note: To select an element shape for meshing, select "Mesh->Controls" from the main menu bar.

Fig. 2.23: Element Type.

way to submit a job as in Ex. 2.2 but within CAE, thus obviating the use of a DOS shell. Note however that execution is a process done outside CAE. CAE maintains a model database (.mdb), and on `Submit`*, it generates an input file (.inp) that is read by the* Abaqus *processor. Although the processor checks the file for errors, it cannot find all possible modeling errors until actual execution. Since, the error check capability of the processor is much less intensive than the solution itself, a* `Data Check` *is normally requested from the Job Manager before requesting a solution. The response is displayed in the information window located at the bottom of the CAE window. If the data check passes, one can submit the job. Again, progress is shown in the information window. Once the solution is completed, hopefully without errors, the results are written by the processor in the output database (.odb). At this point one can invoke the Visualization module directly from the Job window by clicking* `Results`*. This last step will change the module to Visualization, read the .odb, and display a preliminary visualization. The process is summarized as follows:*

```
Module: Job
Menu:   Job, Manager,   # opens the Job Manager window
        Create, Name [Job-1], Model: [Model-1],
            Description: [User's heading goes here], Cont, OK
        Data check      # watch the execution window below the WS
        Submit          # watch the execution window below the WS
        Results
```

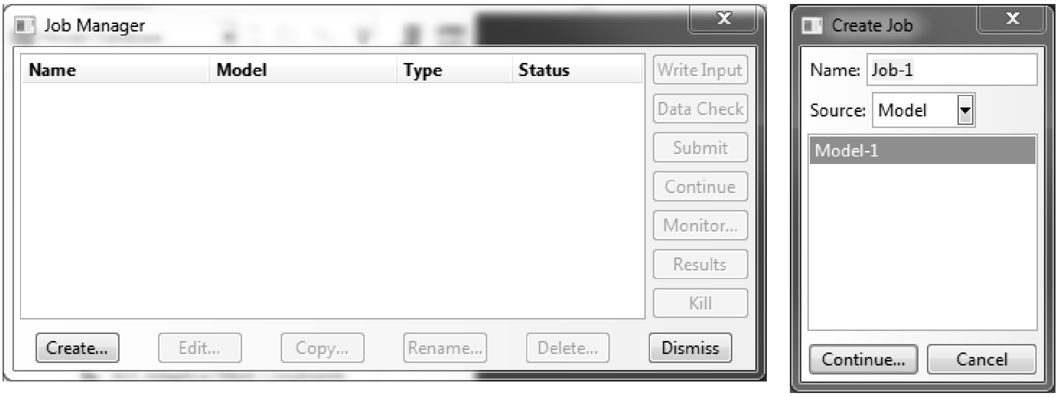


Abaqus/CAE 6.10-2 - Model Database: C:\SIMULIA\User\Chapter_2\Ex_2.3\Ex_2.3.cae [Viewport: 1]

Module: Mesh Model: Model-1 Object: ● Assembly ○ Part:

```
The model database has been saved to "C:\SIMULIA\User\Chapter_2\Ex_2.3\Ex_2.3.cae".
The model database has been saved to "C:\SIMULIA\User\Chapter_2\Ex_2.3\Ex_2.3.cae".
Global seeds have been assigned.
56 elements have been generated on instance: Part-1-1
```

Fig. 2.24: The part is meshed.



Name: Job-1

Source: Model

Fig. 2.25: The Job Manager and Create Job dialog boxes.

The Job Manager dialog and the Create Job dialog boxes are shown in Fig. 2.25. On the Job Manager, note the buttons for Write Input, Data Check, Submit, Continue, Monitor, Results, and Kill. These allow the user to control the execution of the Abaqus *processor, and with Results, to jump directly to the visualization mode after the execution is done.*

The ***Description*** *field allows the user to enter a heading that will display on the visualization module provided* ***Viewport Annotation Options****,* ***General****,* ***Show title box*** *is checked. The heading appears in the* `.inp` *file following the* keyword `*Heading`*, as shown in Example 2.2, p. 47.*

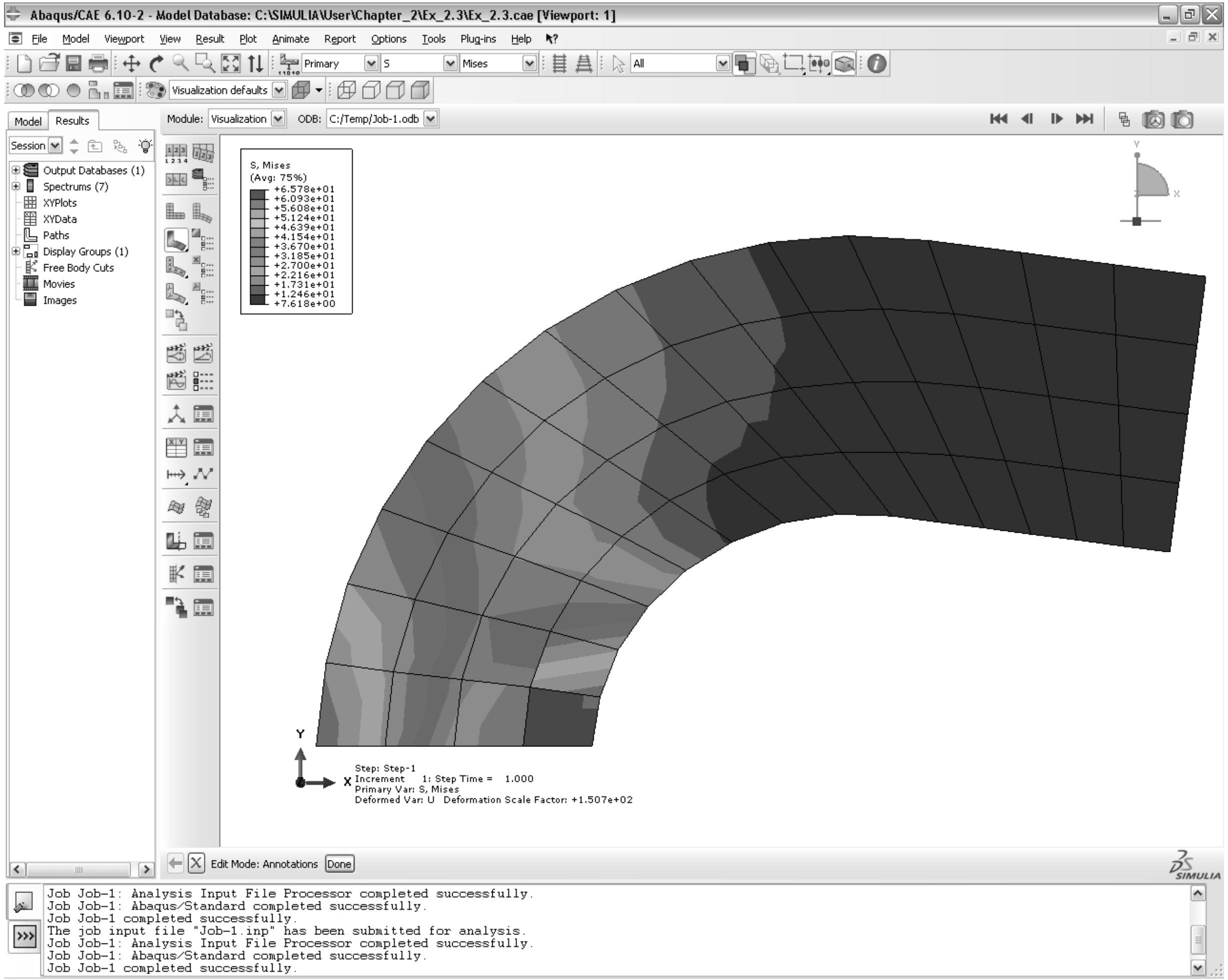
2.3.9 Post-processing and Visualization

Once the solution has been calculated, the post-processor can be used to review and to analyze the results. Results can be reviewed graphically or by listing the values numerically. Since a model usually contains a considerable amount of results, it may be better to use graphical tools. Post-processors of commercial codes produce contour plots of stress and strain distributions, deformed shapes, etc. The software usually includes derived calculations such as error estimation, load case combinations, or path operations.

The visualization module can be reached either from the Job Manager (button Results in Fig. 2.25) or by changing the Module to Visualization using the drop-down Module menu, but the latter requires the user to further instruct CAE to read the .odb by executing `File`, `Open`, `File Filter, [.odb]`, `File Name, [job1.odb]`. *Assuming you reached the visualization module by clicking Results in the Job Manager, a basic visualization can be done as follows:*

```
Module: Visualization
Menu: Plot, Contours, On Deformed Shape
```

This will produce a contour plot of von Mises stress on the deformed shape, as shown in Fig. 2.26. The maximum value of stress is 65.78 MPa*. Finally, before*



```
Job Job-1: Analysis Input File Processor completed successfully.
Job Job-1: Abaqus/Standard completed successfully.
Job Job-1 completed successfully.
The job input file "Job-1.inp" has been submitted for analysis.
Job Job-1: Analysis Input File Processor completed successfully.
Job Job-1: Abaqus/Standard completed successfully.
Job Job-1 completed successfully.
```

Fig. 2.26: Visualize a contour plot of von Mises stress on the deformed shape.

exiting the program, save the various files that contain the status of Abaqus/*CAE at this point in the session; as follows:*

```
Menu: File, Save   # save the .cae and .mdb files
Menu: File, Exit   # exit CAE
```

The `.mdb` *file contains all the model information. The file* `.cae` *allows you to resume the session at a later time. A copy of* ***Ex_ 2. 3. cae*** *is available in the Web site [5].*

Examples 2.4 and 2.5 include commands to review the results by listing and by graphic output, respectively.

Example 2.4 *([8, Example 2.5])Use a commercial FE code to find the axial displacement at the axially loaded end of a rod clamped at the other end. The rod is made of steel* $E = 200{,}000\ MPa$*, diameter* $d = 9\ mm$*, length* $L = 750\ mm$*, and load* $P = 100{,}000\ N$*. Also find the stress and strain. Use three two-node (linear)* ***truss*** *elements (called* ***link*** *elements in Ansys).*

Solution to Example 2.4 *The* Abaqus/*CAE procedure is listed below. The CAE file is available in [5, Ex_2.4.cae]. Poisson's ratio should not be needed for a 1D analysis of a bar, but it is required in* Abaqus/*CAE for defining the isotropic material, so the value* $\nu = 0.3$ *is used.*

i. *Creating the geometry*

```
Module: Part
Menu:   Create, Name: [Part-1], 2D, Deformable, Wire,
    Approx size: [2000], Cont
Menu:   Add, Line, Connected Lines,
    # enter coordinates in the dialog box below the WS
    [0,0], Enter,
    [750,0], Enter, X, Done
```

ii. *Entering the material and section properties*

```
Module: Property
Menu:   Material, Create, Name [Material-1],
    Mechanical, Elasticity, Elastic, Isotropic, [200E3, 0.3], OK
Menu:   Section, Create, Name: [Section-1], Beam, Truss, Cont
    Material-1, Cross-sectional area: [63.617], OK
Menu:   Assign, Section, # pick the truss, Done, OK
```

iii. *Creating an assembly, which is trivial in this case*

```
Module: Assembly
Menu:   Instance, Create, [Part-1], Independent, OK
```

iv. *Creating a step as part of the solution strategy*

```
Module: Step
Menu:   Step, Create, Name [Step-1], Static, General, Cont, OK
```

v. *Specifying loads and boundary conditions, as well as adjusting the display*

```
Module: Load
Menu:   BC, Create, Name: [BC-1],
    Step: Initial, Category: Mechanical, Types: Symm, Cont
    # pick the origin, Done,
    Encastre, OK
Menu:   Load, Create, Name: [Load-1],
    Step: Step-1, Category: Mechanical, Types: Concentrated, Cont
    # pick the end @ x=750, Done,
    CF1 [100E3], OK
Menu:   View, Assembly display options,
    Attribute, Symbol, Size: [24], Arrows: [24], Apply, OK
    F6  # roll the mouse wheel forward to see the part smaller
```

vi. *Meshing*

```
Module: Mesh
Menu:   Seed, Instance, Apply, # to see the proposed mesh seeds
    # change it to get 3 elements
    Approx global size [250], Apply, OK
Menu:   Mesh, Element Type, Family: Truss, Linear, OK
Menu:   Mesh, Instance, Yes
Menu:   View, Assembly display options,
    Tab: Mesh, # checkmark Show node labels, Apply
    # checkmark Show element labels, Apply, OK
    # to save these options as default for the future
Menu: File, Save Options, Current, OK
```

vii. *Defining a Job for the Abaqus processor to execute*

```
Module: Job
Menu:   Job, Manager,
    Create, Name [Job-1], Cont, OK
    Submit,  # monitor the progress window below the WS
    Results, # switches to Module: Visualization
```

viii. *Visualizing the results is easier by invoking the Results from the Job Manager as in the previous step. Once in the Visualization module*

```
Module: Visualization
    # to get arrows with magnitude proportional to the displacement
    # max. value read on the color scale is 5.895 mm
Menu:   Result, Field Output,
    Tab: Symbol Variable, Name: U, Vector: Resultant, Apply, OK
    # to get the stress
    # max. value read on the color scale is 1572 MPa
Menu:   Result, Field Output, Tab: Primary Variable,
    Name: S, Component: S11, Apply, Contour, OK
    # to get the strain
    # max. value read on the color scale 7860E-3 (micro-strain)
Menu:   Result, Field Output, Tab: Primary Variable,
    Name: E, Component: E11, Apply, Contour, OK
    # further modify the way things are seen on the screen
    # and save the settings
```

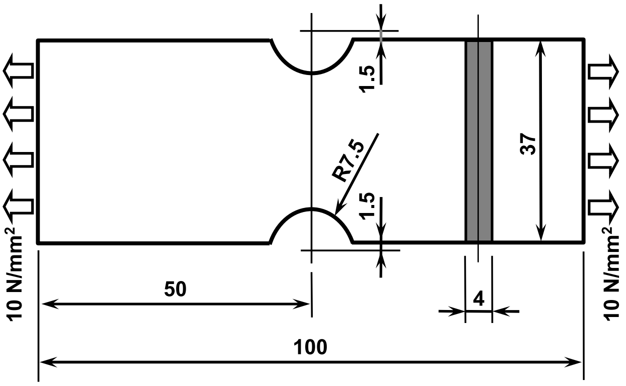


Fig. 2.27: Rectangular notched strap analyzed in Example 2.5.

```
Menu:   View, ODB Display Options, Tab: Entity Display,
            Symbol size [24]
Menu:   File, Save Options, Current, OK
```

A convenient combination of units for this case is Newton, mm, and $MPa = N/mm^2$*. The analysis results can be easily verified by mechanics of material calculations, as follows*

$$U_x = \frac{PL}{AE} = \frac{(750)(100000)}{(63.617)(200000)} = 5.894\ mm$$

$$\sigma = \frac{P}{A} = \frac{100000}{63.617} = 1571.9\ MPa$$

$$\epsilon = \frac{\sigma}{E} = 7.859 \cdot 10^{-3}$$

Example 2.5 *[8, Example 2.6]Use a commercial FE code to find the stress concentration factor of a rectangular notched strap. The dimensions and the load state are defined in Figure 2.27. Use eight node (quadratic) quadrilateral plane stress elements.*

Solution to Example 2.5 *The* Abaqus/*CAE procedure is listed below. The CAE file is available in [5, Ex_2.5.cae].*

i. *Creating the geometry*

```
Module: Part,
Menu: Part, Create, Name: [Part-1],
    2D, Deformable, Shell, Approx size [200], Cont
    # a quarter model of the Figure in the textbook
Menu: Add, Point,
    [0,0]
    [50,18.5]
    X,
Menu: Add, Line, Rectangle, #pick diagonal points, X
    Add, Point,
    [0,20]      # center
    [7.5,20]    # radius
    X,
Menu: Add, Circle     # select center and edge point
    Edit, Auto trim # the unneeded lines/arches
    X,
    Done    # with the sketch. finalize the part.
Menu: File, Save
```

ii. *Entering the material and section properties*

```
Module: Property
Menu: Material, Create, Name: [Material-1],
        Mechanical, Elasticity, Elastic, [190E3, 0.3], OK
Menu: Section, Create, Name: [Section-1], Solid, Homogeneous, Cont
       # checkmark Plane stress/strain thickness: [4], OK
Menu: Assign, Section, # pick the part, Done, OK
```

iii. *Creating an assembly*

```
Module: Assembly
Menu: Instance, Create, Name: [Part-1], Independent, OK
```

iv. *Creating a step as part of the solution strategy*

```
Module: Step
Menu: Step, Create, Name: [Step-1], Static, General, Cont, OK
```

v. *Specifying loads and boundary conditions, as well as adjusting the display*

```
Module: Load
Menu:   BC, Create, Name: [BC-1], Step: Initial, Mechanical, Symm, Cont
    # pick the horiz edge on bottom, Done, YSYMM, OK
Menu:   BC, Create, Name: [BC-2], Step: Initial, Mechanical, Symm, Cont
    # pick the vert edge on left, Done, XSYMM, OK
Menu:   Load, Create, Name: [Load-1],
    Step: Step-1, Mechanical, Types: Pressure, Cont
    # pick the vertical edge on the right, Done,
    Distribution: Uniform, Magnitude [-10], OK
    # mouse wheel fwd to zoom out on the WS
```

vi. *Meshing*

```
Module: Mesh
# Above the WS, select Object: Assembly
Menu: Seed, Instance, Apply, #if you like it, OK,
    # otherwise Approx global size: [1.0], Apply, OK
Menu: Mesh, Controls, Quadratic, Structured, OK
Menu: Mesh, Element type, Standard, Quadratic,
        Family: Plane stress, Reduced Integration, OK
Menu: Mesh, Instance, Yes, # OK to mesh the part instance
```

vii. *Defining a Job for the Abaqus processor to execute*

```
Module: Job
Menu:   Job, Manager
    Create, Name: Job-1, Cont, OK
    Submit  # the job for execution by the Abaqus processor
    Results # switches to Module:Visualization
```

viii. *Visualizing the results*

```
Module: Visualization
# to get a report file
Menu:   Report, Field output,
    Tab: Variable, Position: Integ. point, expand S, # checkmark S11
    Tab: Setup, Name: [abaqus.rpt],
    OK, # immediately the .rpt file is written in the current WD
    # min S11 = 0 at intersection of hole and horz symm line
    # max S11 ~ 28.1 at intersection of hole and vert symm line
    # or
    # integration point location varies upon mesh refinement
    # the node with max stress is always the same
    # unique nodal
    # displays average stress from neighbor elements at node
    # element nodal also, but displays the element number as well
    Tab: Variable, Position: Unique nodal, expand S, check S11
    Tab: Setup, Name: [abaqus.rpt], OK
    # open abaqus.rpt with notepad++
    # min S11 = 0 @ node 5
    # max S11 ~ 27.9 @ node 7
    # to see the node #'s
Module: Mesh, Object: Assembly
Menu:   View, Assembly display options,
    Tab: Mesh, # checkmark Show node labels, Apply
```

The stress in the net area without stress concentration is

$$\sigma_o = \frac{P}{A} = \frac{10 \cdot 37 \cdot 4}{25 \cdot 4} = 14.8\ MPa$$

From the FE model, the maximum horizontal stress close to the notch is approximately 27.9 MPa. *Therefore, the stress concentration factor is*

$$k = \frac{\sigma_{max}}{\sigma_o} = 1.885$$

Example 2.6 *[8, Example 2.4]Using* Abaqus/*CAE, generate a model for a dome with different types of elements (shell and beam elements), using different section properties and two materials. By solving this example the user will become familiar with the construction of 3D parts in* Abaqus/*CAE.*

Solution to Example 2.6 *To facilitate manipulating the part during the construction, activate the* `Views toolbar` *as follows:*

```
Menu: View, Toolbars, Views.
# Integrate it next to the other toolbars above the WS area.
```

Creating the 3D dome

i. *Creating a solid semi-sphere:*

```
Module: Part
Menu: Part, Create, Name [Part-1], 3D, Deformable, Solid,
    Revolution, Approximate size [2000], Cont,
Menu: Add, Arc, Center/Endpoint,
    # in the dialog box below the WS enter the coordinates
    [0,0], [0,500], [500,0], X # to end the Arc command, F6 # zoom/fit
Menu: Add, Line, Connected Lines,
    # enter the coordinates
    [0,500], [0,0], [500,0], X # to end the Line command, Done,
    Angle [360], OK
```

A 3D semi-sphere should have been created.

ii. *Creating the side-walls:*

```
# Rotate the part so you can see the base of the semi-sphere,
Menu: View, Rotate, X # to end the Rotate command,
    # roll the mouse-wheel backward to zoom in,
    # save this view-point for use in future procedures,
    # click the 'Save View' icon from the 'Views toolbar',
    # in the pop-up window select [User 1], Save Current, OK,
Menu: Shape, Cut, Extrude, # pick the base of the semi-sphere,
    # pick the edge of the base of the semi-sphere,
    # Sketch mode is now active,
Menu: Add, Line, Rectangle, # enter the coordinates
    [300,300], [-300,-300],
    # create a new rectangle entering the coordinates
    [600,600], [-600,-600], X # to end the Line command, Done,
    Type: [Through All], OK
```

The semi-sphere should have been cut creating four identical side-walls as shown in Figure 2.28.

iii. *Reducing the height of the side-walls:*

```
Menu: Shape, Cut, Extrude, # pick one of the side-walls,
    # in the dialog box below the WS select [horizontal on bottom],
    # pick the longest straight edge of the side-wall selected,
    # Sketch mode is now active,
    # roll the mouse-wheel forward to see the figure smaller,
    # Draw a rectangle in an area away from the figure,
Menu: Add, Line, Rectangle, # pick any two point on the screen,
    X # to end Line command
```

Your construction should look similar to Figure 2.29.

```
Menu: Add, Dimension, # pick a vertical line of the rectangle
    # may need to move pointer and right-click to create dimension
    # enter the height of the rectangle [200],
    # pick a horizontal line of the rectangle,
    # enter the width of the rectangle [600],
    X # to end the Dimension command
```

Your construction should look similar to Figure 2.30.

```
Menu: Edit, Transform, Translate,
    # in the notification area below the WS select [Move],
    # pick the four lines forming the rectangle
    # hold the 'shift-key' to individually select the lines, Done,
    # pick the lower-left point of the rectangle,
    # pick the lower-left point of the figure,
    X # to end the Translate command, Done
```

The rectangle should have been moved as shown in Figure 2.31.

```
    Type: [Through All], OK
```

The height of the part should have been reduced as shown in Figure 2.32.

iv. *Converting the solid part to a shell part:*

```
Menu: Shape, Shell, From Solid,
    # pick the part, Done, X # to end the shell command,
Menu: Tools, Geometry Edit, Face, Remove,
    # pick the surface at the base of the part,
    Done, X # to end the command,  # close the pop-up window
```

The part should look like the one shown in Figure 2.33.

v. *Creating the supporting columns:*

```
Menu: Shape, Wire, Sketch, # pick one of the side-walls,
    # in the dialog box below the WS select [horizontal on bottom],
    # pick the longest straight edge of the side-wall selected,
    # Sketch mode is now active,
    # roll the mouse-wheel forward to see the figure smaller,
Menu: Add, Line, Connected Lines,
    # pick the lower-left corner of the part,
    # pick any point directly below (it must create a vertical line),
    X # to end the Line command,
Menu: Add, Dimension, # pick the line,
    # enter the length of the line [200],
    X # to end the Dimension command,
    # Repeat the procedure to create a vertical line of length 200
    # starting at the lower-right point of the part,
    Done # to end the sketch
```

Create another sketch on the opposite side-wall, and repeat the procedure above to create the two additional vertical columns. The part should look like the one shown in Figure 2.34.

Materials, and sections properties:

```
Module: Property
```

i. *Defining Materials*

```
Menu: Material, Manager,
    Create, Name [Dome], Mechanical, Elasticity, Elastic, Isotropic,
    Young’s Modulus [190e3], Poisson’s ratio [0.27], OK,
    Create, Name [Columns], Mechanical, Elasticity, Elastic, Isotropic,
    Young’s Modulus [200e3], Poisson’s ratio [0.29], OK,
    # close the Material Manager pop-up window
```

ii. *Defining Section Properties*

```
Menu: Section, Manager,
    Create, Name [Dome-Roof], Shell, Homogeneous, Cont,
    Shell thickness [6.0], Material [Dome], OK,
    Create, Name [Side-Walls], Shell, Homogeneous, Cont,
    Shell thickness [4.0], Material [Dome], OK,
    Create, Name [Columns], Beam, Truss, Cont, Material [Columns],
    Cross-sectional area [100], OK,
    # close the Material Manager pop-up window
```

iii. *Assign Sections*

```
Menu: Assign, Section,
    # pick the roof of the dome, Done, Section [Dome-Roof], OK,
    # pick the side-walls,
    # hold the ’shift-key’ to individually select the surfaces,
    Done, Section [Side-Walls], OK,
    # pick the columns,
    # hold the ’shift-key’ to individually select the lines,
    Done, Section [Columns], OK,
    X # to end the command
```

Assembly:

```
Module: Assembly
Menu: Instance, Create, Parts [Part-1], Independent, OK
```

Loads and BC:

i. *Creating a Load and BC Step*

```
Module: Step
Menu: Step, Create, Name [Step-1], General, Static/Gen., Cont, OK,
    # Return to the saved view,
    # click the ’Apply User 1 View’ icon from the ’Views toolbar’
```

ii. *Defining Boundary Conditions*

```
Module: Load
Menu: BC, Manager,
    Create, Name [BC-1], Step: Initial, Mechanical, Sym/Ant/Enc, Cont,
    # pick the lower-end of the four columns,
    # hold the 'shift-key' to individually select the points, Done,
    Encastre, OK,
    Create, Name [BC-2], Step: Initial, Mechanical, Disp/Rota, Cont,
    # pick the vertical edges of the side-walls,
    # hold the 'shift-key' to individually select the sides, Done,
    # checkmark U1, and U3, OK,
    # Close the Boundary Condition Manager pop-up window,
Menu: Load, Create, Name [Load-1], Step: Step-1,
    Mechanical, Pressure, Cont, # pick the dome-roof, Done,
    # in the notification area below the WS select [Brown],
    Distribution: Uniform, Magnitude [-100], OK,
    # click the 'Apply Iso View' icon from the 'Views toolbar',
    # roll the mouse-wheel backward to zoom in,
    # increase the size of the BC and Load markers,
Menu: View, Assembly Display Options,
    Tab: Attribute, Symbol, Size [18], Arrows [18], Face Density [10],
    Apply, OK
```

The part should look similar to the one shown in Figure 2.35.

Meshing the Part:

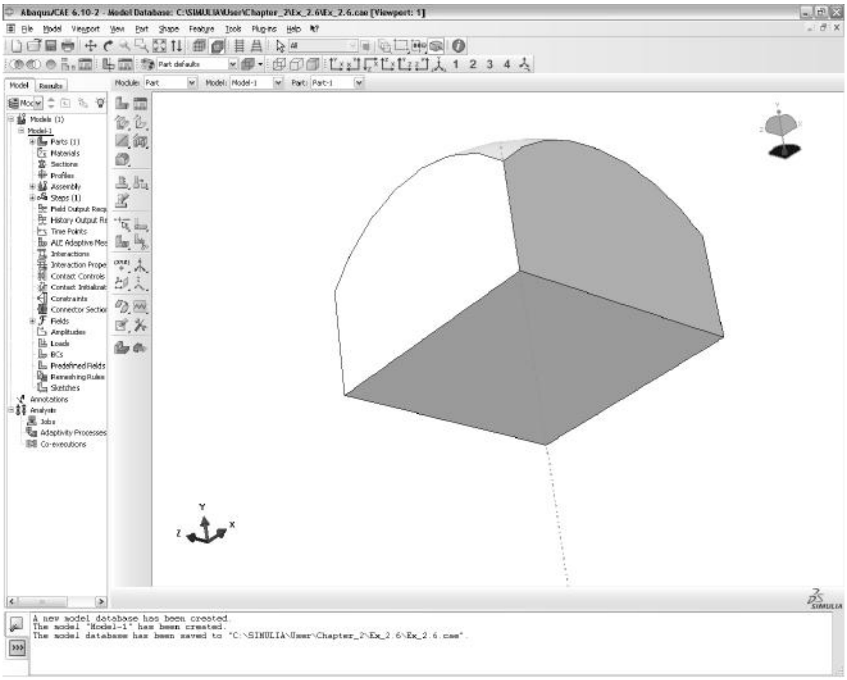
```
# Return to the saved view,
# click the 'Apply User 1 View' icon from the 'Views toolbar',
Module: Mesh
Menu: Seed, Instance, Approximate Global Size [30], Apply, OK,
Menu: Mesh, Element Type,
    # pick the roof of the dome, Done, Standard, Linear, Family [Shell], OK,
    # pick the four side-walls, Done, Standard, Linear, Family [Shell], OK,
    # pick the four columns, Done, Standard, Linear, Family [Truss], OK,
    X # to end the command,
Menu: Mesh, Instance,
    # in the notification area below the WS select [Yes],
    # click the 'Apply Iso View' icon from the 'Views toolbar',
    # roll the mouse-wheel backward to zoom in
```

The part should look similar to the one shown in Figure 2.36.

Running the analysis and visualizing the results:

```
Module: Job
Menu: Job, Manager,
    Create, Name [Job-1], Cont, OK
    Submit, # when the Status message indicates 'Completed'
    Results
Module: Visualization
Menu: Plot, Contours, On Deformed Shape
```

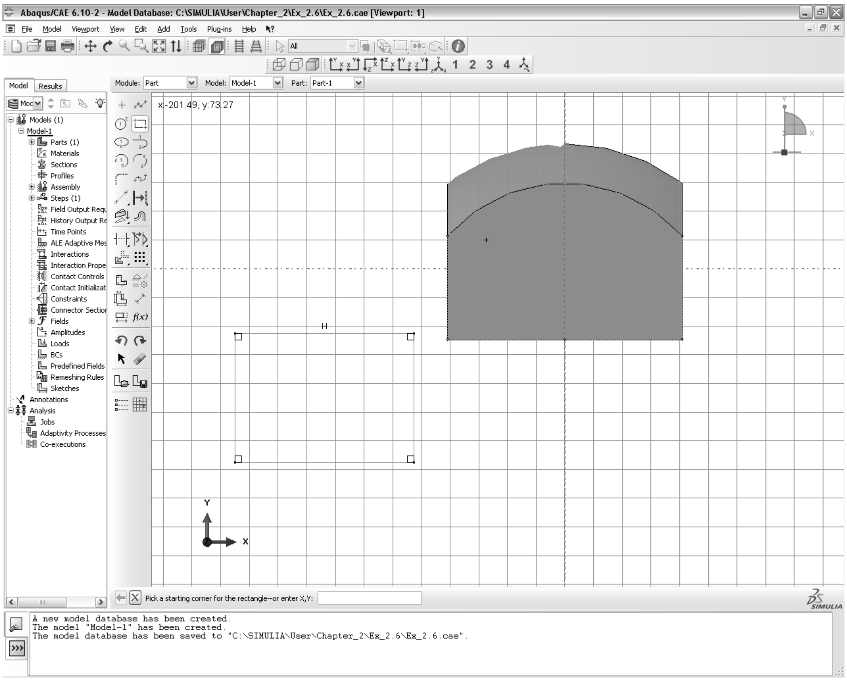
The part should look similar to the one shown in Figure 2.37. The CAE file is available in [5, Ex_2.6.cae].


Abaqus/CAE 6.10-2 - Model Database: C:\SIMULIA\User\Chapter_2\Ex_2.6\Ex_2.6.cae [Viewport: 1]

Module: Part Model: Model-1 Part: Part-1

```
A new model database has been created.
The model "Model-1" has been created.
The model database has been saved to "C:\SIMULIA\User\Chapter_2\Ex_2.6\Ex_2.6.cae".
```

Fig. 2.28: Dome's side-walls.


Abaqus/CAE 6.10-2 - Model Database: C:\SIMULIA\User\Chapter_2\Ex_2.6\Ex_2.6.cae [Viewport: 1]

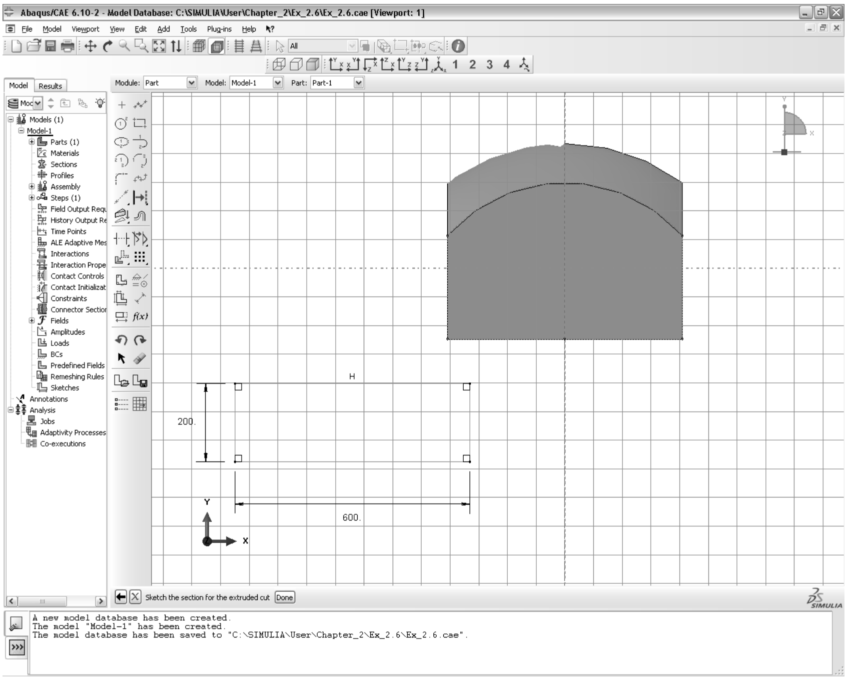
Module: Part Model: Model-1 Part: Part-1

x:-201.49, y:73.27

Pick a starting corner for the rectangle--or enter X,Y:

```
A new model database has been created.
The model "Model-1" has been created.
The model database has been saved to "C:\SIMULIA\User\Chapter_2\Ex_2.6\Ex_2.6.cae".
```

Fig. 2.29: Dome's side-walls height reduction step 1.



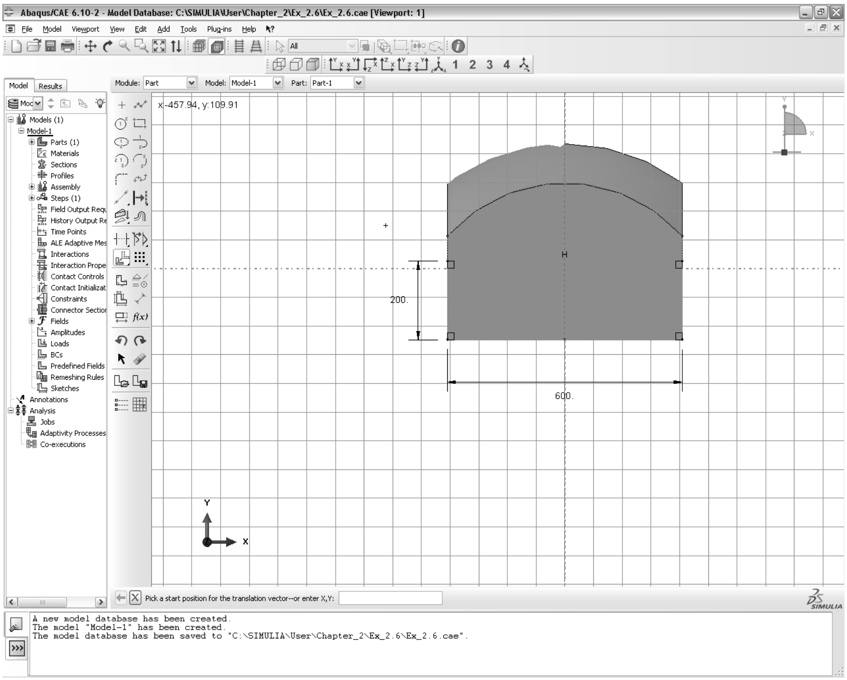
Abaqus/CAE 6.10-2 - Model Database: C:\SIMULIA\User\Chapter_2\Ex_2.6\Ex_2.6.cae [Viewport: 1]

Module: Part Model: Model-1 Part: Part-1

Sketch the section for the extruded cut Done

```
A new model database has been created.
The model "Model-1" has been created.
The model database has been saved to "C:\SIMULIA\User\Chapter_2\Ex_2.6\Ex_2.6.cae".
```

Fig. 2.30: Dome's side-walls height reduction step 2.



Abaqus/CAE 6.10-2 - Model Database: C:\SIMULIA\User\Chapter_2\Ex_2.6\Ex_2.6.cae [Viewport: 1]

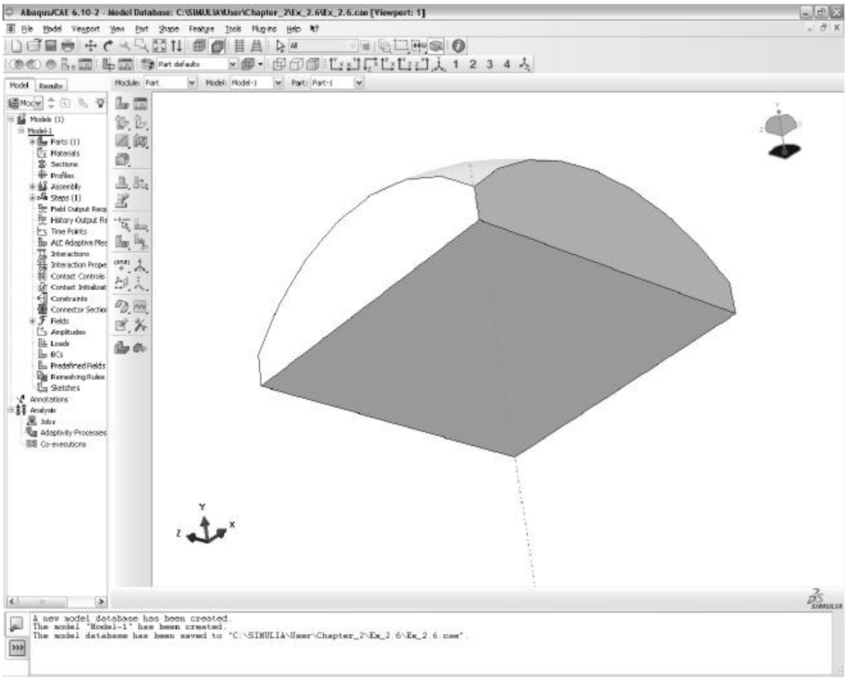
Module: Part Model: Model-1 Part: Part-1

x:-457.94, y:109.91

Pick a start position for the translation vector--or enter X,Y:

```
A new model database has been created.
The model "Model-1" has been created.
The model database has been saved to "C:\SIMULIA\User\Chapter_2\Ex_2.6\Ex_2.6.cae".
```

Fig. 2.31: Dome's side-walls height reduction step 3.

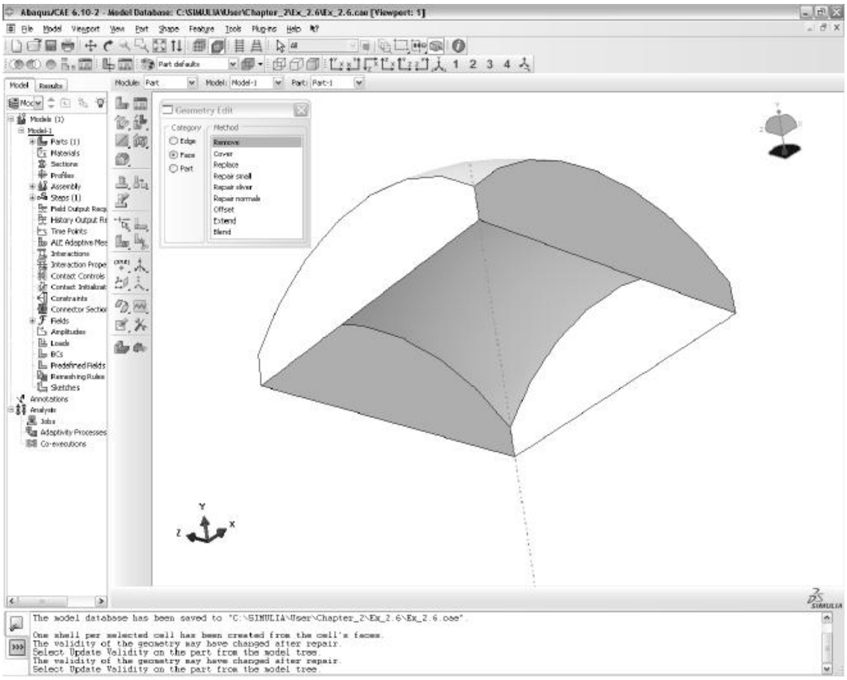


Abaqus/CAE 6.10-2 - Model Database: C:\SIMULIA\User\Chapter_2\Ex_2.6\Ex_2.6.cae [Viewport: 1]

Module: Part Model: Model-1 Part: Part-1

```
A new model database has been created.
The model "Model-1" has been created.
The model database has been saved to "C:\SIMULIA\User\Chapter_2\Ex_2.6\Ex_2.6.cae".
```

Fig. 2.32: Dome's side-walls height reduction step 4.



Abaqus/CAE 6.10-2 - Model Database: C:\SIMULIA\User\Chapter_2\Ex_2.6\Ex_2.6.cae [Viewport: 1]

Module: Part Model: Model-1 Part: Part-1

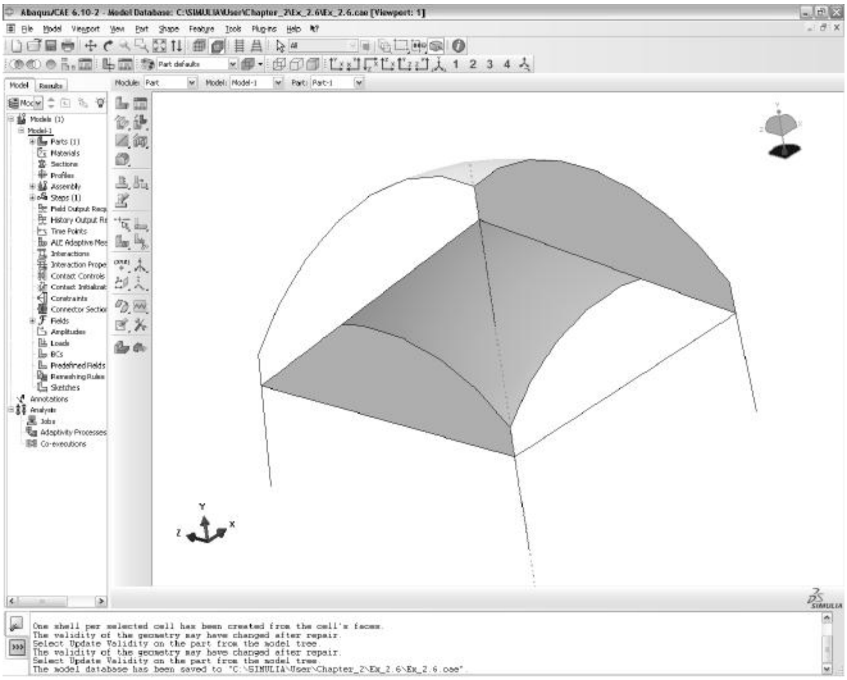
Geometry Edit

Category: ○ Edge ● Face ○ Part

Method: Remove

```
The model database has been saved to "C:\SIMULIA\User\Chapter_2\Ex_2.6\Ex_2.6.cae".

One shell per selected cell has been created from the cell's faces.
The validity of the geometry may have changed after repair.
Select Update Validity on the part from the model tree.
The validity of the geometry may have changed after repair.
Select Update Validity on the part from the model tree.
```

Fig. 2.33: Dome as shell structure.

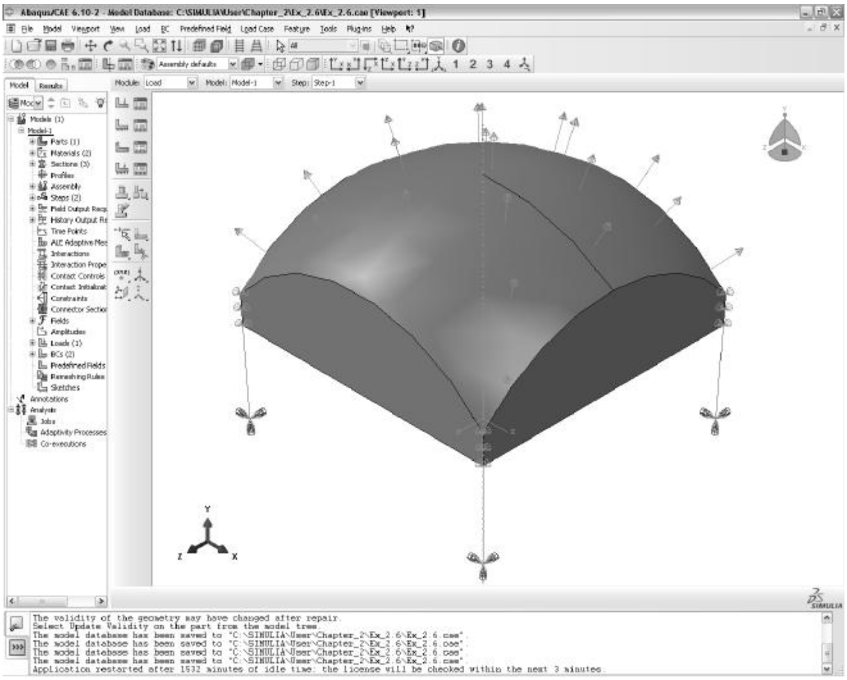


Abaqus/CAE 6.10-2 - Model Database: C:\SIMULIA\User\Chapter_2\Ex_2.6\Ex_2.6.cae [Viewport: 1]

Module: Part Model: Model-1 Part: Part-1

```
One shell per selected cell has been created from the cell's faces.
The validity of the geometry may have changed after repair.
Select Update Validity on the part from the model tree.
The validity of the geometry may have changed after repair.
Select Update Validity on the part from the model tree.
The model database has been saved to "C:\SIMULIA\User\Chapter_2\Ex_2.6\Ex_2.6.cae".
```

Fig. 2.34: Dome completed.

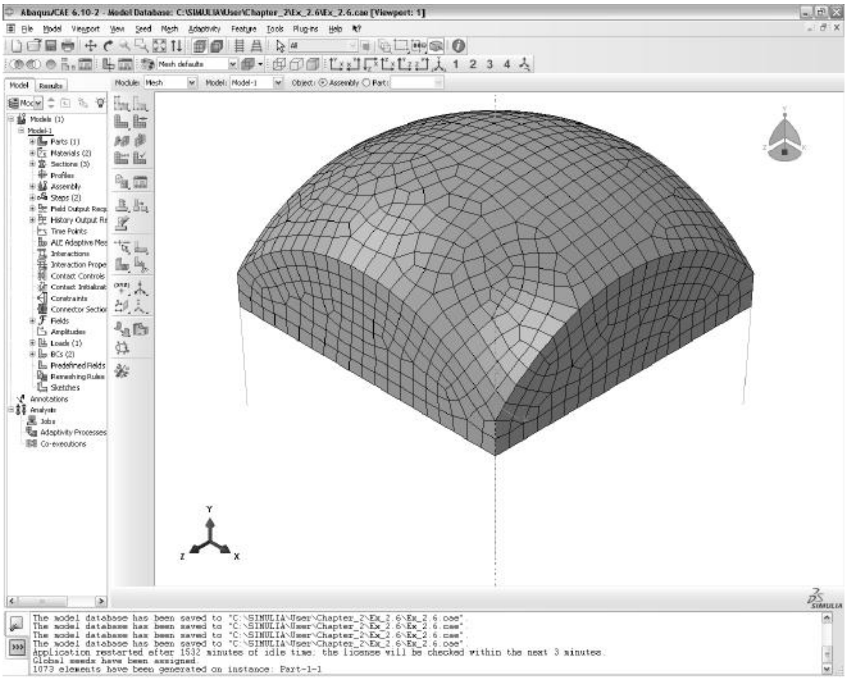


Abaqus/CAE 6.10-2 - Model Database: C:\SIMULIA\User\Chapter_2\Ex_2.6\Ex_2.6.cae [Viewport: 1]

Module: Load Model: Model-1 Step: Step-1

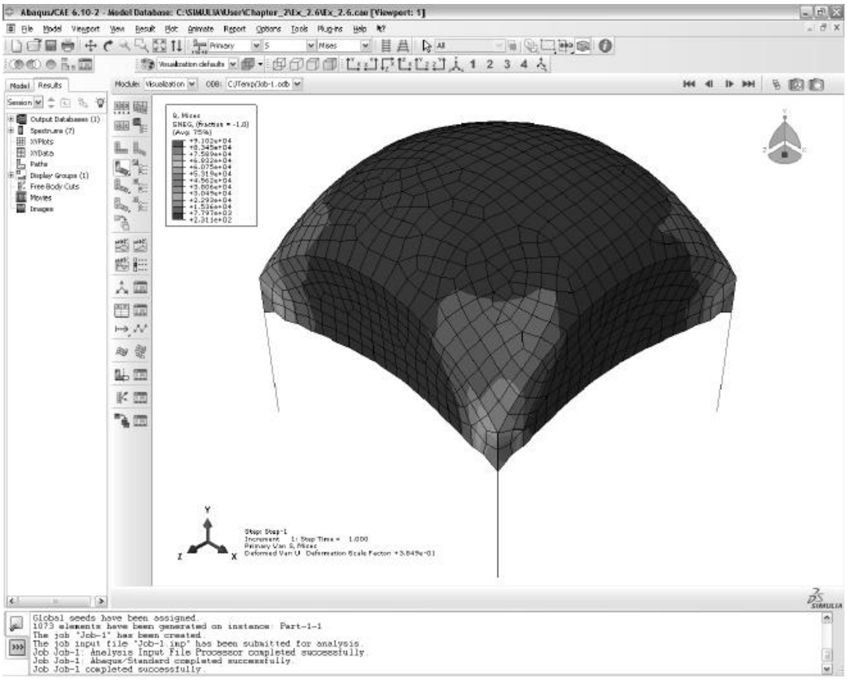
```
The validity of the geometry may have changed after repair.
Select Update Validity on the part from the model tree.
The model database has been saved to "C:\SIMULIA\User\Chapter_2\Ex_2.6\Ex_2.6.cae".
The model database has been saved to "C:\SIMULIA\User\Chapter_2\Ex_2.6\Ex_2.6.cae".
The model database has been saved to "C:\SIMULIA\User\Chapter_2\Ex_2.6\Ex_2.6.cae".
The model database has been saved to "C:\SIMULIA\User\Chapter_2\Ex_2.6\Ex_2.6.cae".
Application restarted after 1532 minutes of idle time: the license will be checked within the next 3 minutes.
```

Fig. 2.35: Loaded Dome structure.


Abaqus/CAE 6.10-2 - Model Database: C:\SIMULIA\User\Chapter_2\Ex_2.6\Ex_2.6.cae [Viewport: 1]
Module: Mesh
Model: Model-1
Object: Assembly Part:
Y
Z
X

```
The model database has been saved to "C:\SIMULIA\User\Chapter_2\Ex_2.6\Ex_2.6.cae".
The model database has been saved to "C:\SIMULIA\User\Chapter_2\Ex_2.6\Ex_2.6.cae".
The model database has been saved to "C:\SIMULIA\User\Chapter_2\Ex_2.6\Ex_2.6.cae".
The model database has been saved to "C:\SIMULIA\User\Chapter_2\Ex_2.6\Ex_2.6.cae".
Application restarted after 1532 minutes of idle time: the license will be checked within the next 3 minutes.
Global seeds have been assigned.
1073 elements have been generated on instance: Part-1-1
```

Fig. 2.36: Meshed Dome structure.



```
Global seeds have been assigned.
1073 elements have been generated on instance: Part-1-1
The job "Job-1" has been created.
The job input file "Job-1.inp" has been submitted for analysis.
Job Job-1: Analysis Input File Processor completed successfully.
Job Job-1: Abaqus/Standard completed successfully.
Job Job-1 completed successfully.
```

Fig. 2.37: Contour plot of von Mises stress. Maximum values are $91,020\ MPa$ in the columns and $66,922\ MPa$ in the shell.

Suggested Problems

Problem 2.1 *Solve Example 2.4 (p. 75) explicitly as it is done in Section 2.1, using only two elements. Show all work.*

Problem 2.2 *From the solution of Problem 2.1, compute the axial displacement at (a)* $x = 500$ *mm, (b)* $x = 700$ *mm.*

Problem 2.3 *Using the same procedure in Example 2.1 (p. 46) calculate the element stiffness matrix and the equivalent force vector of a three-node element rod with quadratic interpolation functions. The interpolation functions are*

$$N_1^e = \frac{x - x_2}{x_1 - x_2}\frac{x - x_3}{x_1 - x_3} \qquad N_2^e = \frac{x - x_3}{x_2 - x_3}\frac{x - x_1}{x_2 - x_1} \qquad N_3^e = \frac{x - x_1}{x_3 - x_1}\frac{x - x_2}{x_3 - x_2}$$

where x_1*,* x_2*, and* x_3 *are the coordinate positions of node 1, 2, and 3 respectively. Use* $x_1 = 0$*,* $x_2 = h/2$*, and* $x_3 = h$*, where* h *is the element length. Show all work.*

Problem 2.4 *Program a FE code using the element formulation obtained in Example 2.1 (p. 46) and the assembly procedure shown in Section 2.1.6. With this code, solve Example 2.4, p. 75. Show all work in a report.*

Problem 2.5 *Program a FE code using the element formulation obtained in Problem 2.3 and the assembly procedure shown in Section 2.1.6. With this code, solve Example 2.4, p. 75. Show all work in a report.*

Chapter 3

Elasticity and Strength of Laminates

Most composite structures are built as assemblies of plates and shells. This is because the structure is more efficient when it carries membrane loads. Another important reason is that thick laminates are difficult to produce.

For example, consider a beam made of an homogeneous material with tensile and compressive strength σ_u subjected to bending moment M. Further, consider a solid beam of square cross section (Figure 3.1), equal width and depth $2c$, with area A, inertia I, and section modulus S given by

$$\begin{aligned} A &= 4c^2 \\ I &= \frac{4}{3}c^4 \\ S &= \frac{I}{c} = \frac{4}{3}c^3 \end{aligned} \tag{3.1}$$

When the stress on the surface of the beam reaches the failure stress σ_u, the bending moment per unit area is

$$m_u = \frac{M_u}{A} = \frac{S\sigma_u}{A} = \frac{1}{3}c\sigma_u \tag{3.2}$$

Now consider a square hollow tube (Figure 3.1) of dimensions $2c \times 2c$ and wall thickness t, with $2c >> t$, so that the following approximations are valid

$$\begin{aligned} A &= 4(2c)t = 8ct \\ I &= 2\left[\frac{t(2c)^3}{12} + c^2(2ct)\right] = \frac{16}{3}tc^3 \\ S &= \frac{I}{c} = \frac{16}{3}tc^2 \end{aligned} \tag{3.3}$$

Then

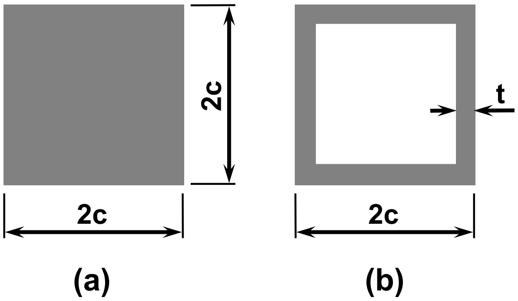


Fig. 3.1: Solid section (a) and hollow square tube (b).

$$m_u = \frac{M_u}{A} = \frac{S\sigma_u}{A} = \frac{\frac{16}{3}tc^2\sigma_u}{8ct} = \frac{2}{3}c\sigma_u \tag{3.4}$$

The failure moment per unit area m_u is twice as large for a hollow square tube with thin walls than for a solid section.

Of course, the failure moment is limited by buckling of the thin walls (see Chapter 4). This is the reason buckling analysis is so important for composites. Most composite structures are designed under buckling constraints because the thicknesses are small and the material is very strong; so normally one does not encounter material failure as in metallic structures (e.g., yield stress) but structural failure such as buckling.

Plates are a particular case of shells, having no initial curvature. Therefore, only shells will be mentioned in the sequel. Shells are modeled as two-dimensional structures because two dimensions (length and width) are much larger than thickness. The thickness coordinate is eliminated from the governing equations so that the 3D problem simplifies to 2D. In the process, the thickness becomes a parameter that is known and supplied to the model.

Modeling of laminated composites differs from modeling conventional materials in three aspects. First, the constitutive equations of each lamina are orthotropic (Section 1.12.3). Second, the constitutive equations of the element depend on the kinematic assumptions of the shell theory used and their implementation into the element. Finally, material symmetry is as important as geometric and load symmetry when trying to use symmetry conditions in the models.

3.1 Kinematic of Shells

Shell elements are based on various shell theories, which in turn are based on kinematic assumptions. That is, there are some underlying assumptions about the likely type of deformation of the material. These assumptions are needed to reduce the 3D governing equations to 2D. Such assumptions are more or less appropriate for various situations, as discussed next.

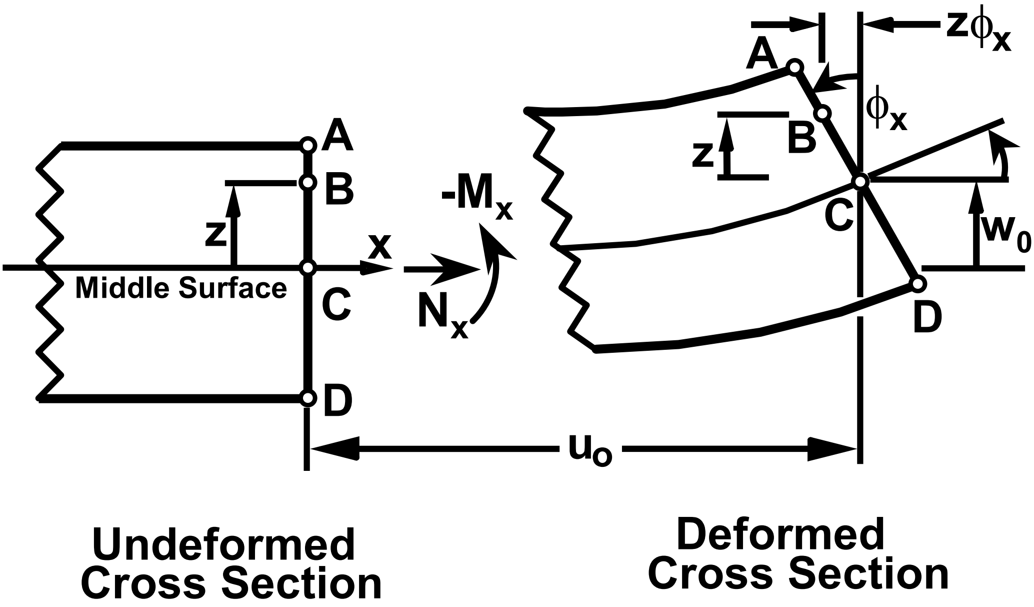


Fig. 3.2: Assumed deformation in FSDT.[1]

3.1.1 First-Order Shear Deformation Theory

The most popular composite shell theory is the first-order shear deformation theory (FSDT). It is based on the following assumptions:

i. A straight line drawn through the thickness of the shell in the undeformed configuration may rotate but it will remain straight when the shell deforms. The angles it forms (if any) with the normal to the undeformed midsurface are denoted by ϕ_x and ϕ_y when measured in the $x - z$ and $y - z$ planes, respectively (Figures 2.19 and 3.2).

ii. The change of the shell thickness is negligible as the shell deforms.

These assumptions are verified by experimental observation in most laminated shells when the following are true:

- The aspect ratio $r = a/t$, defined as the ratio between the shortest surface dimension a and the thickness t, is larger than 10.

- The stiffness of the laminas in shell coordinates (x, y, z) does not differ by more than two orders of magnitude. This restriction effectively rules out sandwich shells, where the core is much softer than the faces.

[1] Reprinted from *Introduction to Composite Materials Design*, E. J. Barbero, Fig. 6.2, copyright (1999), with permission from Taylor & Francis.

Based on the assumptions above, the displacement of a generic point B anywhere in the shell can be written in terms of the displacement and rotations at the midsurface C as

$$\begin{aligned} u(x,y,z) &= u_0(x,y) - z\phi_x(x,y) \\ v(x,y,z) &= v_0(x,y) - z\phi_y(x,y) \\ w(x,y,z) &= w_0(x,y) \end{aligned} \tag{3.5}$$

The midsurface variables on the right-hand side of (3.5) are functions of only two coordinates (x and y); thus the shell theory is 2D. On the left-hand side, the displacements are functions of three coordinates, and thus correspond to the 3D representation of the material. At the 3D level, we use the 3D constitutive equations (1.68) and the 3D strain-displacement equations (1.5), which now can be written in terms of 2D quantities as

$$\begin{aligned} \epsilon_x(x,y,z) &= \frac{\partial u_0}{\partial x} - z\frac{\partial \phi_x}{\partial x} = \epsilon_x^0 + z\kappa_x \\ \epsilon_y(x,y,z) &= \frac{\partial v_0}{\partial y} - z\frac{\partial \phi_y}{\partial y} = \epsilon_y^0 + z\kappa_y \\ \gamma_{xy}(x,y,z) &= \frac{\partial u_0}{\partial y} + \frac{\partial v_0}{\partial x} - z\left(\frac{\partial \phi_x}{\partial y} + \frac{\partial \phi_y}{\partial x}\right) = \gamma_{xy}^0 + z\kappa_{xy} \\ \gamma_{yz}(x,y) &= -\phi_y + \frac{\partial w_0}{\partial y} \\ \gamma_{xz}(x,y) &= -\phi_x + \frac{\partial w_0}{\partial x} \\ \epsilon_z &= 0 \end{aligned} \tag{3.6}$$

where

- The midsurface strains ϵ_x^0, ϵ_y^0 , γ_{xy}^0, also called membrane strains, represent stretching and in-plane shear of the midsurface.

- The change in curvature κ_x, κ_y, κ_{xy}, which are close but not exactly the same as the geometric curvatures of the midsurface. They are exactly that for the Kirchhoff theory discussed in Section 3.1.2.

- The intralaminar shear strains γ_{xz}, γ_{yz}, which are through-the-thickness shear deformations. These are small but not negligible for laminated composites because the intralaminar shear moduli G_{23}, G_{13} are small when compared with the in-plane modulus E_1. Metals are relatively stiff in shear ($G = E/2(1+\nu)$), and thus the intralaminar strains are negligible. In addition, the intralaminar shear strength values F_4, F_5, are relatively small when compared to the in-plane strength values F_{1t}, F_{1c}, thus making evaluation of intralaminar strains (and possibly stresses) a necessity. On the other hand, the shear strength

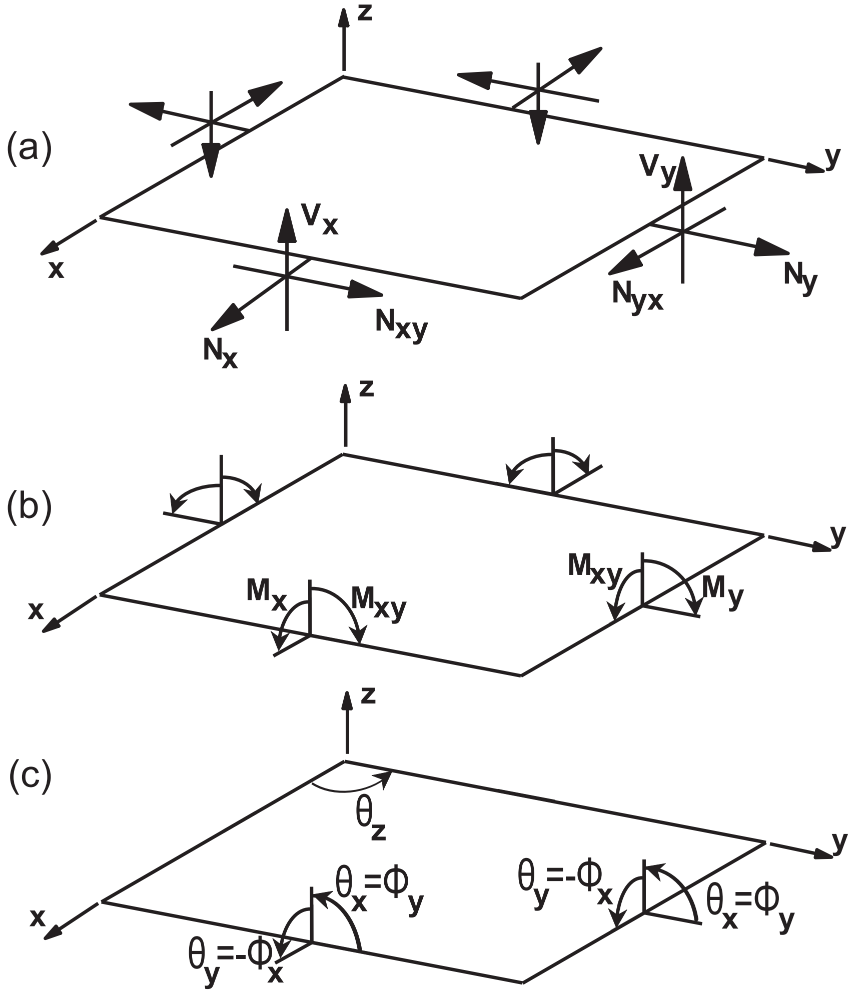


Fig. 3.3: Stress resultants acting on a plate or shell element: (a) forces per unit length, (b) moments per unit length, and (c) definition of shell theory rotations ϕ compared to mathematical angles θ.

of metals is comparable to their tensile strength, and since the intralaminar stress is always smaller than the in-plane stress, it is not necessary to check for intralaminar failure of metallic homogeneous shells. That is not the case for laminated metallic shells since the adhesive is not quite strong and it may fail by intralaminar shear.

While the 3D constitutive equations relate strains to stress, the laminate constitutive equations relate midsurface strains and curvatures. The laminate constitutive equations are obtained by using the definition of stress resultants. While in 3D elasticity every material point is under stress, a shell is loaded by stress resultants (Figure 3.3), which are simply integrals of the stress components through the thickness of the shell, as follows

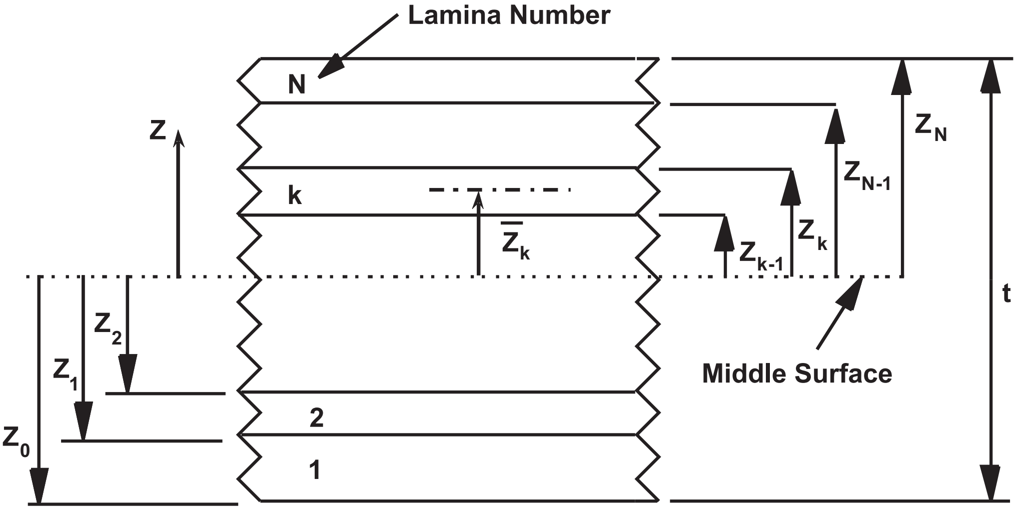


Fig. 3.4: Definition of the z-coordinate system to locate the interface between laminas (z_k) and the midsurface of the laminas ($\overline{z}_k$)

$$\left\{\begin{array}{c} N_x \\ N_y \\ N_{xy} \end{array}\right\} = \sum_{k=1}^{N} \int_{z_{k-1}}^{z_k} \left\{\begin{array}{c} \sigma_x \\ \sigma_y \\ \sigma_{xy} \end{array}\right\}^k dz$$

$$\left\{\begin{array}{c} V_y \\ V_x \end{array}\right\} = \sum_{k=1}^{N} \int_{z_{k-1}}^{z_k} \left\{\begin{array}{c} \sigma_{yz} \\ \sigma_{xz} \end{array}\right\}^k dz$$

$$\left\{\begin{array}{c} M_x \\ M_y \\ M_{xy} \end{array}\right\} = \sum_{k=1}^{N} \int_{z_{k-1}}^{z_k} \left\{\begin{array}{c} \sigma_x \\ \sigma_y \\ \sigma_{xy} \end{array}\right\}^k z\ dz \tag{3.7}$$

where N is the number of layers, z_{k-1} and z_k are the coordinates at the bottom and top surfaces of the k-th layer, respectively (Figure 3.4). Replacing the plane stress version of the 3D constitutive equations in shell local coordinates (1.100-1.101) at each layer and performing the integration we get

$$\left\{\begin{array}{c} N_x \\ N_y \\ N_{xy} \\ M_x \\ M_y \\ M_{xy} \end{array}\right\} = \left[\begin{array}{cccccc} A_{11} & A_{12} & A_{16} & B_{11} & B_{12} & B_{16} \\ A_{12} & A_{22} & A_{26} & B_{12} & B_{22} & B_{26} \\ A_{16} & A_{26} & A_{66} & B_{16} & B_{26} & B_{66} \\ B_{11} & B_{12} & B_{16} & D_{11} & D_{12} & D_{16} \\ B_{12} & B_{22} & B_{26} & D_{12} & D_{22} & D_{26} \\ B_{16} & B_{26} & B_{66} & D_{16} & D_{26} & D_{66} \end{array}\right] \left\{\begin{array}{c} \epsilon_x^0 \\ \epsilon_y^0 \\ \gamma_{xy}^0 \\ \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{array}\right\} \tag{3.8}$$

$$\left\{\begin{array}{c} V_y \\ V_x \end{array}\right\} = \left[\begin{array}{cc} H_{44} & H_{45} \\ H_{45} & H_{55} \end{array}\right] \left\{\begin{array}{c} \gamma_{yz} \\ \gamma_{xz} \end{array}\right\}$$

where

$$\begin{aligned}
A_{ij} &= \sum_{k=1}^{N} \left(\overline{Q}_{ij}\right)_k t_k; \qquad i,j = 1,2,6 \\
B_{ij} &= \sum_{k=1}^{N} \left(\overline{Q}_{ij}\right)_k t_k \bar{z}_k; \qquad i,j = 1,2,6 \\
D_{ij} &= \sum_{k=1}^{N} \left(\overline{Q}_{ij}\right)_k \left(t_k \bar{z}_k^2 + \frac{t_k^3}{12}\right); \qquad i,j = 1,2,6 \\
H_{ij} &= \frac{5}{4} \sum_{k=1}^{N} \left(\overline{Q}_{ij}^*\right)_k \left[t_k - \frac{4}{t^2}\left(t_k \bar{z}_k^2 + \frac{t_k^3}{12}\right)\right]; \qquad i,j = 4,5
\end{aligned} \tag{3.9}$$

where $\left(\overline{Q}_{ij}\right)_k$ are the coefficients in laminate coordinates of the plane-stress stiffness matrix for layer number k, t_k is the thickness of layer k, and $\bar{z}_k$ is the coordinate of the middle surface of the kth layer. For an in-depth discussion of the meaning of various terms see [1]. In summary, the A_{ij} coefficients represent in-plane stiffness of the laminate, the D_{ij} coefficients represent bending stiffness, the B_{ij} represent bending-extension coupling, and the H_{ij} represent intralaminar shear stiffness. All these coefficients can be calculated by (3.9) and are implemented in widely available software packages such as CADEC [12].

When membrane and bending deformations are uncoupled (e.g., symmetric laminates), the governing equations of FSDT involve three variables for solving the bending problem (w^0, ϕ_x, ϕ_y) and two to solve the membrane problem (u^0, v^0). Bending-extension coupling means that all five variables will have to be found simultaneously, which is what FEA software codes do for every case, whether the problem is coupled or not.

The equilibrium equations of plates can be derived by using the PVW (see (1.16)). Furthermore, the governing equations can be derived by substituting the constitutive equations (3.8) into the equilibrium equations.

3.1.2 Kirchhoff Theory

Historically, Kirchhoff theory was preferred because the governing equations can be written in terms of only one variable, the transverse deflection of the shell w_0. In the pre-information age, it was easier to obtain analytical solutions in terms of only one variable rather than in terms of the three variables needed in FSDT. This means that a wealth of closed form design equations and approximate solutions exist in engineering design manuals which are based on Kirchhoff theory [13]. Such simple design formulas can still be used for preliminary design of composite shells if we are careful and we understand their limitations. Metallic shells were and still are commonly modeled with Kirchhoff theory. The FSDT governing equations can be reduced to Kirchhoff governing equations, and closed form solutions can be found, as shown in [14].

In Kirchhoff theory the intralaminar shear strain is assumed to be zero. From the last two equations in (3.6) we get

$$
\begin{aligned}
\phi_x &= \frac{\partial w_0}{\partial x} \\
\phi_y &= \frac{\partial w_0}{\partial y}
\end{aligned}
\tag{3.10}
$$

and introducing them into the first three equations in (3.6) we get

$$
\begin{aligned}
\epsilon_x(x,y,z) &= \frac{\partial u_0}{\partial x} - z\frac{\partial^2 w_0}{\partial x^2} = \epsilon_x^0 + z\kappa_x \\
\epsilon_y(x,y,z) &= \frac{\partial v_0}{\partial y} - z\frac{\partial^2 w_0}{\partial y^2} = \epsilon_y^0 + z\kappa_y \\
\gamma_{xy}(x,y,z) &= \frac{\partial u_0}{\partial y} + \frac{\partial v_0}{\partial x} - 2z\frac{\partial^2 w_0}{\partial x \partial y} = \gamma_{xy}^0 + z\kappa_{xy}
\end{aligned}
\tag{3.11}
$$

Notice that the variables ϕ_x, ϕ_y have been eliminated and Kirchhoff theory only uses three variables $u_0(x,y)$, $v_0(x,y)$, and $w_0(x,y)$. This makes analytical solutions easier to find, but numerically Kirchhoff theory is more complex to implement. Since second derivatives of w_0 are needed to write the strains, the weak form (2.30) will have second derivatives of w_0. This will require that the interpolation functions (see Section 2.1.4) have C^1 continuity. That is, the interpolation functions must be such that not only the displacements but also the slopes be continuous across element boundaries. In other words, both the displacement w_0 and the slopes $\partial w_0/\partial x$, $\partial w_0/\partial y$ will have to be identical at the boundary between elements when calculated from either element sharing the boundary. This is difficult to implement.

Consider the case of beam bending. The ordinary differential equation (ODE) with an applied distributed load $\widehat{q}(x)$ is

$$
EI\frac{d^4 w_0}{dx^4} = \widehat{q}(x)
\tag{3.12}
$$

The weak form is obtained as in (2.3)

$$
0 = \int_{x_A}^{x_B} v\left[-EI\frac{d^4 w_0}{dx^4} + \widehat{q}(x)\right] dx
\tag{3.13}
$$

Integrating by parts twice

$$
\begin{aligned}
0 &= \int_{x_A}^{x_B} \frac{d^2 v}{dx^2} EI \frac{d^2 w_0}{dx^2} dx + vEI\frac{d^3 w_0}{dx^3} - \frac{dv}{dx}EI\frac{d^2 w_0}{dx^2} - \int_{x_A}^{x_B} v\widehat{q}(x)dx \\
0 &= B(v,w_0) + [vQ_x]_{x_A}^{x_B} - \left[\frac{dv}{dx}M_x\right]_{x_A}^{x_B} - \int_{x_A}^{x_B} v\widehat{M}(x)dx \\
0 &= B(v,w_0) + L(v)
\end{aligned}
\tag{3.14}
$$

When the elements are assembled as in Section 2.1.6, it turns out that adjacent elements i and $i+1$ that share a node have identical deflection but opposite shear force Q_x and bending moment M_x at their common node, as follows

$$\begin{aligned} w^i &= w^{i+1} \\ Q^i &= -Q^{i+1} \\ M^i &= -M^{i+1} \end{aligned} \tag{3.15}$$

For the shear forces to cancel as in (2.24), it is only required to have $v^i = v^{i+1}$, which is satisfied by C^0 continuity elements having $w^i = w^{i+1}$ at the common node. For the bending moments to cancel as in (2.24), it is required that $dw^i/dx = dw^{i+1}/dx$. This can only be done if the elements have C^1 continuity. That is, the slopes $dw^i/dx = dw^{i+1}/dx$ must be identical at the common node. Such elements are difficult to work with ([15, p. 276]).

In FSDT theory, only first derivatives are used in the strains (3.6). So, the weak form (2.30) has only first derivatives and, like (2.24), all the internal generalized forces cancel at common nodes with only C^0 element continuity.

3.1.3 Simply Supported Boundary Conditions

Composite plates with coupling effects may have bending, shear, and membrane deformations coupled even if loaded by pure bending, pure shear, or pure in-plane loads (see [1, Figure 6.7]). While the term *simply supported* always means to restrict the transverse deflection $w(x,y)$, it does not uniquely define the boundary conditions on the in-plane displacements u_n and u_s, normal and tangent to the boundary, respectively. In the context of analytical solutions, it is customary to restrict either u_n or u_s. Therefore, the following possibilities exist

– SS-1: $w = u_s = \phi_s = 0 \quad ; \quad N_n = \widehat{N}_n \quad ; \quad M_n = \widehat{M}_n$

– SS-2: $w = u_n = \phi_s = 0 \quad ; \quad N_{ns} = \widehat{N}_{ns} \quad ; \quad M_n = \widehat{M_n}$

In type SS-1, a normal force and a moment are specified. In SS-2, a shear force and a moment are specified. The naming convention for the rotations is the same as that used for moment resultants in Figure 3.3, where a subscript $()_n$ indicates the direction normal to the edge of the shell, and a subscript $()_s$ indicates the direction tangent to the edge (see also [14, Figure 6.2.1]). Note that the rotation vector ϕ_s is perpendicular to the direction s, resulting in $\phi_s \approx \frac{\partial w}{\partial s}$; therefore, both SS-1 and SS-2 set that rotation to zero. Finally, $\widehat{()}$ represents a fixed known value, which may or may not be zero.

Geometrically linear analysis of a symmetric laminate subjected to bending only will not develop noticeable u_n, u_s, displacements. Therefore, SS-1 and SS-2 should give virtually identical results. Differences may be important for other cases. Furthermore, the condition $\phi_s = 0$ should not be neglected. For example, the calculated center deflection was found to increase by 11.5% when the condition $\phi_s = 0$ was not included in the solution of a sandwich plate [16] using 36 S4R elements.

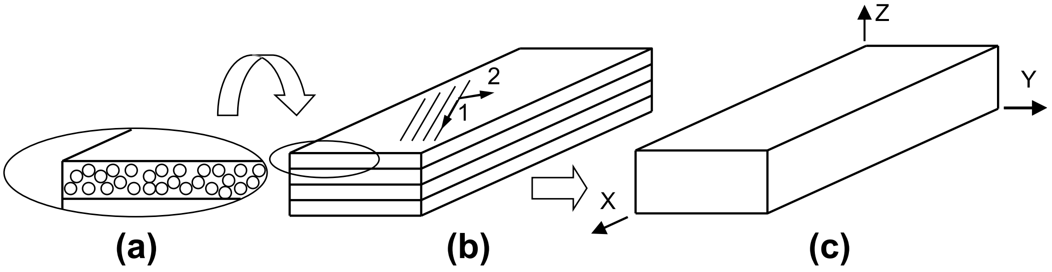


Fig. 3.5: (a) Micromechanics, (b) lamina level, and (c) laminate level approach.

3.2 Finite Element Analysis of Laminates

Deformation and stress analysis of laminated composites can be done at different levels (Figure 3.5). The level of detail necessary for description of the material depends on the level of post-processing desired.

When a great level of detail is necessary (Figure 3.5.a), the strain and stress are computed at the constituent level, i.e. fiber and matrix. In this case, it is necessary to describe the microstructure, including the fiber shape and geometrical distribution, and the material properties of the constituents. More details are given in Chapter 6 where micromechanical modeling is used to generate properties for any combination of fibers and matrix. Also, when the composite material is a woven fabric, or the laminate is very thick, or when studying localized phenomena such as free edges effects, the composite should be analyzed as solid, as shown in Chapter 5. However, it must be noted that most of the laminated structures can be analyzed using the plates and shell simplifications explained in Section 3.1.

At the other end of the spectrum (Figure 3.5.c), the composite material can be considered as a homogeneous equivalent material. In this case, its structural behavior can be analyzed by using orthotropic properties shown in Chapter 1. If the whole laminate is analyzed as a homogeneous equivalent shell, using the macroscale level approach (Figure 3.5.c), the stress distribution in the laminate cannot be obtained. However, this very simple description of the laminate is sufficient when only displacements, buckling loads and modes, or vibration frequencies and modes are required. In these cases, only the laminate stiffness (3.8) is needed (see Section 3.2.9). In certain cases, even a simpler material description will suffice. For example, when the laminate is only unidirectional, or if the laminate is balanced and symmetric (see [1, Section 6.3]), the laminate can be modeled as a single lamina of orthotropic material (Section 3.2.10).

In most cases, stress and strains need to be calculated for every lamina in the laminate. Then, the actual laminate stacking sequence (LSS) must be input to the program (Section 3.2.11). In this case, the elastic properties of each lamina, as well as thickness and fiber orientation of every lamina, must be given. This method is usually called the mesoscale level approach (Figure 3.5.b).

A unidirectional lamina can be satisfactorily approximated as being transversely isotropic. Then, it suffices to use $E_3 = E_2$, and $G_{23} = E_3/2(1 + \nu_{23})$ in the equations for an orthotropic material. The elastic properties of a unidirectional lamina can be computed using micromechanics (Chapter 6) or with experimental data of unidirectional laminates. Material properties of some unidirectional composites are shown in Table 3.1.

In the analysis of most composite structures, it is usual to avoid the micromechanics approach and to obtain experimentally the properties of the unidirectional lamina, or even the whole laminate. However, the experimental approach is not ideal because a change of constituents or fiber volume fraction during the design process invalidates all the material data and requires a new experimental program for the new material. It is better to calculate the elastic properties of the lamina using micromechanics formulas, using software such as [12] (see also Section 6.1). Unfortunately, micromechanics formulas are not accurate to predict strength, so experimental work cannot be ruled out completely.

In summary, laminate properties can be specified in two ways:

- by the constitutive matrices A, B, D, and H, or
- by specifying the laminate stacking sequence (LSS) and properties for every lamina.

When the constitutive matrices A, B, D, H of the laminate are used to define the laminate, the shell element cannot distinguish between different laminas. It can only relate generalized forces and moments to generalized strains and curvatures. On the other hand, layered shell elements have the capability to compute the laminate properties using the *laminate stacking sequence* (LSS) and the properties of the laminas.

3.2.1 Element Types and Naming Convention

Shell elements allow one to model thin to moderately thick shells, down to a side-to-thickness ratio of 10. While some of them have 3 or 4 nodes, others have 8 nodes, thus using interpolation functions of higher degree. Shell elements are defined in 3D space and have 5 or 6 degrees of freedom (DOF) at each node (translations in the nodal x-, y-, and z-directions and rotations about the nodal x-, y-, and z-axis). The 6th DOF (rotation about the z axis) is included in the shell formulation to allow modeling of folded plates, but it would not be necessary if the shell surface is smooth.

The *Abaqus* documentation uses several terms that at first may seem confusing and/or overlapping. We have summarized them as follows:

Conventional shell elements require the geometry (mesh) to represent a 2D flat/curved surface in 3D, as opposed to continuum shell elements. Conventional shell elements include thin shell elements, such as S8R5, and thick shell elements such as S8R.

Continuum shell elements require the geometry to model explicitly the thickness of the shell, as it would be done with 3D solid elements, but in contrast

Table 3.1: Material properties of unidirectional carbon/epoxy composites.[a]

Property	Unit	AS4D/9310	T300/5208
E_1	[GPa]	133.86	136.00
$E_2 = E_3$	[GPa]	7.706	9.80
$G_{12} = G_{13}$	[GPa]	4.306	4.70
G_{23}	[GPa]	2.76	4.261
$\nu_{12} = \nu_{13}$		0.301	0.280
ν_{23}		0.396	0.150
V_f		0.55	
ρ	[g/cm^3]	1.52	1.54
α_1	[$10^{-6}/°C$]	0.32	
α_2	[$10^{-6}/°C$]	25.89	
F_{1t}	[MPa]	1830	1550
F_{1c}	[MPa]	1096	1090
$F_{2t} = F_{3t}$	[MPa]	57	59
$F_{2c} = F_{3c}$	[MPa]	228	207
F_4	[MPa]	141	128
F_6	[MPa]	71	75

[a] F_4 was calculated with $\alpha_0 = 54°$ in [1, Eq. 4.109]

to the later, continuum shell elements enforce the FSDT constrains (Section 3.1.1) through particular element interpolation functions. Continuum shell elements include SC6R and SC8R.

Thin shell elements enforce the Kirchhoff constraint (Section 3.1.2), either theoretically (e.g., STRI3), or numerically when the shell is thin (e.g., S8R5). Therefore, the transverse shear deformations are assumed to be zero or negligible. They are not accurate for composites if the laminate is thick and/or the transverse shear moduli G_{23} of one or more laminas are small, in which case the shear deformation may be underestimated. All thin shell elements are conventional shell elements. Thin shell elements include STRI3, S4R5, STRI65, S8R5, S9R5, SAXA1N, and SAXA2N. They do not have a drilling rotation degree of freedom.

Thick shell elements only enforce the FSDT constraint (Section 3.1.1). Therefore, the transverse shear deformations are not zero. Thick shell elements can be conventional (e.g., S3R, S4, S4R, S8R, SAX1, SAX2), or continuum, such as SC6R and SC8R.

General purpose shell elements are appropriate to model both thick and thin shells. They include thick, conventional shell elements (e.g., S3, S3R, S3RS, S4, S4R, S4RS, S8R, S4RSW, SAX1, SAX2, SAX2T), as well as thick, continuum shell elements such as SC6R and SC8R.

3D solid elements also called *solid 3D continuum elements* discretize the 3D body without using any assumptions of shell theory. Since they are not shell elements, the model may be computationally expensive. They can be used for detailed analysis in regions where a rapid variation of stress and strain is expected. They require a 3D mesh, explicitly modeling the 3 dimensions of the shell, including the thickness. Shell-to-solid coupling constraints can be used to transition from shell to solid elements.

Another source of confusion arises around the use of the terms 2D and 3D. In shell theory, a shell is a 2D surface because only 2 curvilinear coordinates are needed to locate any point on the reference surface of the shell. But in *Abaqus* documentation, a shell is said to be 3D because it occupies a portion of 3D space, as opposed to a 2D model that is always planar.

Furthermore, the *Abaqus* documentation uses the following naming convention for shell elements:

SnRsW Conventional Shell

S shell

n number of nodes

R reduced integration

s small membrane strains

W warping included

SCnRT Continuum Shell

S shell

C continuum

n number of nodes

R reduced integration

T thermo-mechanical coupling

STRInm Triangular Shell

S shell

TRI triangular

n number of nodes

m number of DOF

Type SAXAxN Axisymmetric Shell

S shell

AXA Axisymmetric

1N meridional linear interpolation, N Fourier modes

2N meridional quadratic interpolation, N Fourier modes

The terms introduced in this section are further explained in the following sections.

3.2.2 Thin (Kirchhoff) Shell Elements

When a shell is thin and the material has a high shear modulus, lines normal to the middle surface remain normal during deformation. This is one of the assumptions of Kirchhoff shell theory, also called the Kirchhoff constraint (see Section 3.1.2).

The Kirchhoff constraint states that a line normal to the reference surface in the undeformed configuration remains normal to the deformed reference surface. This implies that the transverse shear deformations γ_{xz}, γ_{yz} are zero. Additional assumptions that are common to both Kirchhoff Theory and First Order Deformation Theory (FSDT, Section (3.2.4)) are that normals remain straight and inextensible ($\epsilon_{zz} = 0$). Shell elements that enforce the Kirchhoff constraint are called *thin shell elements*, as opposed to *thick shell elements* that only enforce the FSDT constrains. All the thin shell elements are *conventional shell elements* in the sense that the geometry is represented by a 2D flat/curved surface in 3D, as opposed to *continuum shell elements* that require the geometry to model explicitly the thickness of the shell, as it would be done with 3D solid elements.

Thin shell elements such as STRI3, S4R5, STRI65, S8R5, S9R5, SAXA1N, and SAXA2N are specifically formulated for modeling thin shells. They discretize the reference surface of the shell, and the thickness is given as a section property. Element STRI3 enforces the Kirchhoff constraint analytically so it yields the thin-walled theory solution even for a thick shell. Elements S4R5, STRI65, S8R5, S9R5, SAXA1N, and SAXA2N impose the Kirchhoff constraint numerically. These elements should not be used for applications in which transverse shear deformation is important, either because the shell is thick, or the shear modulus low, or both. See Example 3.4.a, p. 121.

All elements with name ending on 5 have 5 DOF per node, 3 displacements, and 2 in-plane rotations. They do not have drilling rotational DOF (see Section 3.2.8). Five degrees of freedom elements, such as S4R5, STRI65, S8R5, S9R5, may be more economical than other elements, but they are available only for modeling thin shells and they should not be used to model thick shells.

3.2.3 Thick Shell Elements

Composites should be modeled as thick shells even if they are geometrically thin because they have relatively low shear modulus, thus necessitating inclusion of their shear deformation in the same way as thick shells do. *Abaqus* provides elements S3R, S3RS, S4, S4R, S4RS, S4RSW, S8R, SC6R, and SC8R to model laminated composite shells. Continuum shell elements, such as SC6R and SC8R, can be used when better resolution of the transverse shear deformation is needed. See Example 3.4.b, p. 121.

3.2.4 General-purpose (FSDT) Shell Elements

First-Order Shear Deformation Theory (FSDT, see Section 3.1.1) states that undeformed normals to the reference surface in the undeformed configuration remain straight and inextensible but not necessarily normal to the deformed reference surface. This allows for non-zero transverse shear deformation.

General-purpose shell elements, such as S3, S3R, S3RS, S4, S4R, S4RS, S8R, S4RSW, SAX1, SAX2, and SAX2T, as well as SC6R and SC8R, include transverse shear deformation but they can be used for thin shells as well. Elements S3, S3R, S3RS, S4, S4R, S4RS, S4RSW, S8R, SAX1, SAX2, and SAX2T discretize the reference surface of the shell, and the thickness is given as a section property. Elements S3, S3R, S3RS, S4, S4R, S4RS, S4RSW, and S8R have 6 DOF per node, with DOF 6 being a drilling rotation that allows the element to be used to model folded shells. Elements SC6R and SC8R are continuum elements (see Section 3.2.5), but they qualify as general-purpose because they can be used for thin and thick shells.

3.2.5 Continuum Shell Elements

Continuum shell (CS) elements such as SC4R are basically 3D solid elements where the FSDT constraints are enforced by special interpolation functions [17]. Compared to conventional shell elements, they have more nodes to depict the thickness in the model.

Continuum shell elements have only displacement degrees of freedom, no rotations. Therefore, they can be stacked and they can be connected to solid elements. Continuum shell elements discretize the 3D geometry of the shell in the same way as 3D solid elements do.

Continuum shell elements can be internally laminated to represent a laminate or sublaminate with just one element. Also, they can be stacked to provide a better representation of the shear deformation through the thickness. In this case, each of the stacked elements may represent a sublaminate. If only one internally laminated element is used to model the entire thickness of the laminate, the result is the same as that of a FSDT element. But CS elements can be stacked, and there lies their advantage. By stacking several CS elements, the actual deformation of the normal can be approximated to any precision. As the number of elements through the thickness approaches infinity, they yield the exact elasticity solution to the plate problem. But they are better than 3D solid elements because CS elements have no aspect ratio problems; i.e., the thickness direction can be very thin compared to the other two dimensions. This is because CS elements incorporate the incompressibility of the normals ($\epsilon_z = 0$,(3.6)), just like FSDT elements.

Meshing is more difficult with continuum shell elements because the thickness of the shell has to be meshed. Also, boundary conditions are more difficult to apply because one cannot apply the classical shell boundary conditions, but rather one has to simulate them by constraining the displacements at the top and bottom surface of the shell. For example, a simply supported boundary condition is rather difficult to apply if the continuum shell mesh does not have nodes located at the middle surface of the shell. Clamped (encastre) and symmetry boundary conditions are easy to apply. See Example 3.4.c, p. 126.

3.2.6 Sandwich Shells

For a sandwich shell, the core is much softer that the faces, and the transverse shear deformations are significant regardless of the total thickness of the shell. Conventional shell elements may not have the shear flexibility required. Use continuum shell elements instead. For the most refined model possible, stack three continuum shell elements through the thickness, one for each face and the core. See Example 3.7, p. 139.

3.2.7 Nodes and Curvature

Three and four node elements such as STRI3 and S4R5 are flat. For curved shells, it is better to use six, eight, or nine node elements such as STRI65, S8R5, and S9R5. Element S8R5 has a hidden internal ninth-node, which may end up outside the reference surface for doubly curved shells. If this happens, buckling loads may be inaccurate and S9R5 would be better.

3.2.8 Drilling Rotation

Conventional shell elements are based on shell theory, which constrains the 3D continuum deformations according to some kinematic assumptions, such as Kirchhoff, FSDT, or one of many others. In continuum theory, the deformation of a point in 3D can be described in terms of the relative displacements of two points, thus requiring six degrees of freedom (three per point). Both Kirchhoff and FSDT theories formally reduce the requirement to three displacements and two in-plane rotations at a single point. The two rotations, ϕ_x, ϕ_y, are rotations of the normal to the reference surface. These are called in-plane rotations because the rotation vectors lie on the surface of the shell. For a smoothly curved or flat shell, there is no need of tracking the rotation of the normal around itself, ϕ_z. This last rotation is called drilling rotation. However, if the shell has a fold, an in-plane rotation on one side of the fold corresponds to a drilling rotation on the other side (Figure 3.6). Thus, to enforce compatibility of displacements, the drilling rotations become necessary. Since elements having 5 DOF do not have drilling rotation, they are suited to model smoothly curved shells but not folded shells.

Abaqus/Standard will automatically switch from 5 DOF elements to 6 DOF elements at any node when:

- A kinematic (displacement, velocity) boundary condition is applied to rotational degrees of freedom at the node,
- The node is used in a multi-point constraint that involves rotational degrees of freedom,
- The node is shared with a beam element or a shell element that uses the three global rotation components at all nodes,
- The node is on a fold line in the shell, or
- The node is loaded with moments

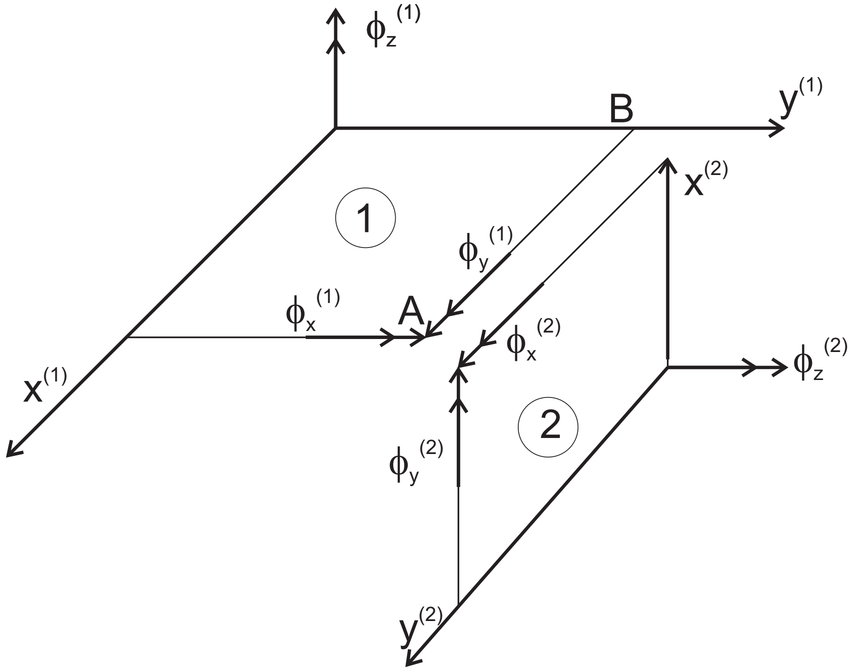


Fig. 3.6: Exploded view of a plate folded along AB. Drilling rotations $\phi_z^{(1)}$ and $\phi_z^{(2)}$ associated to elements 1 and 2, respectively.

3.2.9 A, B, D, H Input Data for Laminate FEA

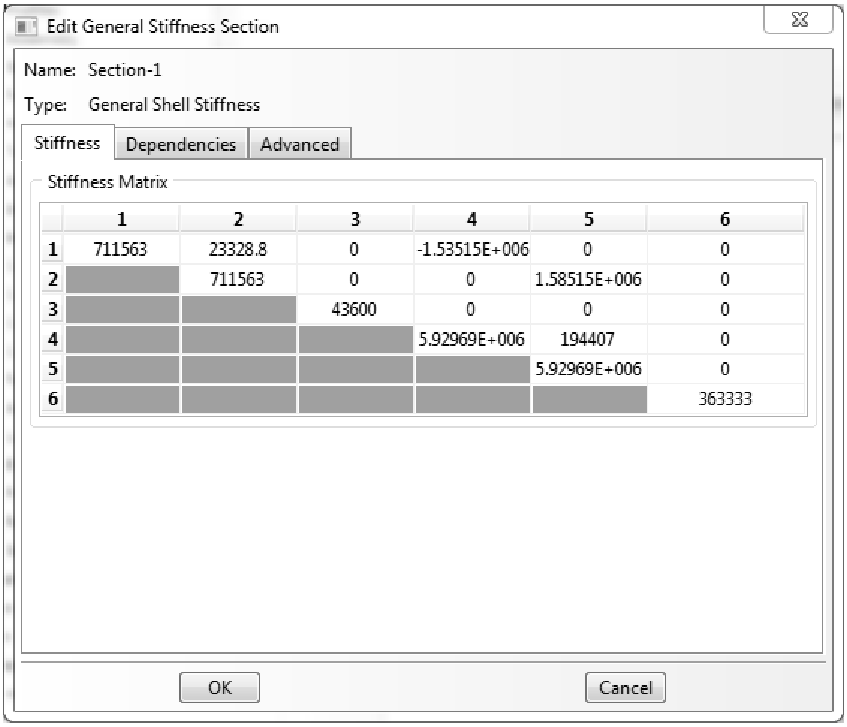
As previously mentioned, macro-scale level (laminate level) analysis is adequate if only deflections, modal analysis, or buckling analysis are to be performed, with no requirement for detailed stress analysis. Then, it is not necessary to specify the laminate stacking sequence (LSS), the thickness, and the elastic properties of each lamina of the laminate. Only the elastic laminate properties (A, B, D, H matrices) defined in (3.9) are required. This is convenient because it allows one to input the aggregate composite material behavior with few parameters. The reduction of the complexity of the input data allows modeling of laminates with an unlimited number of laminas, using only four matrices.

When the A, B, D, H matrices are used to define the FE analysis, the computer model knows the correct stiffness but it does not know the LSS. Therefore, the software can compute the deformation response (including buckling and vibrations) and even the strain distribution through the thickness of the shell, but it cannot compute the stress components because it does not know where the lamina material properties change from lamina to lamina.

The A, B, D, H input data can be found by using (3.9) or [12]. Then, these are input into the FE software, as illustrated in Example 3.1.

In *Abaqus*, the A, B, D matrices can be input as `Edit General Stiffness Section` (Figure 3.7), which is invoked as follows:

```
Module: Property
```



Name: Section-1

Type: General Shell Stiffness

Stiffness Matrix

	1	2	3	4	5	6
1	711563	23328.8	0	-1.53515E+006	0	0
2		711563	0	0	1.58515E+006	0
3			43600	0	0	0
4				5.92969E+006	194407	0
5					5.92969E+006	0
6						363333

Fig. 3.7: Input A, B, and D matrices.

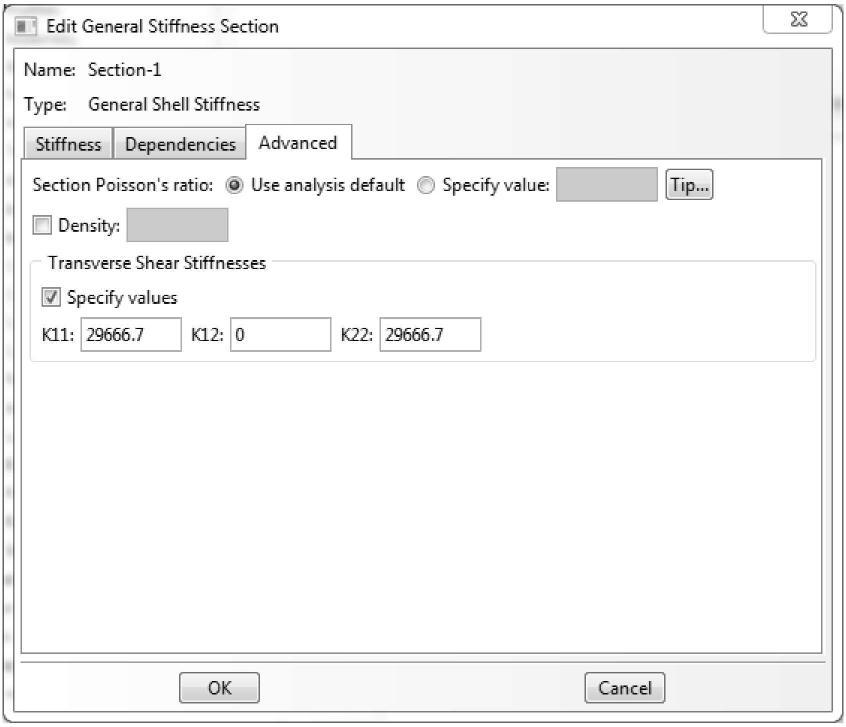
```
Menu:   Section, Edit, [Section-name]
```

The A, B, D, matrices are entered on the `Stiffness` tab (Figure 3.7). The H matrix has to be converted into transverse shear stiffness values and entered using the `Advanced` tab (Figure 3.8). The transverse shear stiffness coefficients H_{44}, H_{45}, H_{55}, are defined in (3.9). Taking into account the way *Abaqus* handles contracted notation, as shown in Table 1.1 (p. 5), compared to the standard (Voigt) definition used in this textbook, it turns out that the notation for transverse shear coefficients in Figure 3.8 means that:

$$\begin{aligned} K_{11} &= H_{55} \\ K_{22} &= H_{44} \\ K_{12} &= H_{45} \end{aligned} \tag{3.16}$$

Example 3.1 *Consider a simply supported square plate $a_x = a_y = 2000$ mm, laminated with AS4D/9310 (Table 3.1) in a $[0/90]_n$ configuration. The total laminate thickness is $t = 10$ mm for all values of n. The plate is loaded in compression with an edge load $N_x = 1$ N/mm and ($N_y = N_{xy} = M_x = M_y = M_{xy} = 0$). Tabulate the center deflection perpendicular to the plate surface when the number of layers is n = 1, 5, 10, 15, and 20. Calculate the A, B, D, H matrices and enter them into* Abaqus. *Use symmetry to model 1/4 of the plate.*

Solution to Example 3.1 *The procedure to solve the $[0/90]$ laminate ($n = 1$) is described in detail next.*



Edit General Stiffness Section

Name: Section-1

Type: General Shell Stiffness

Section Poisson's ratio: ● Use analysis default ○ Specify value: Tip...

□ Density:

Transverse Shear Stiffnesses

☑ Specify values

K11: 29666.7 K12: 0 K22: 29666.7

Fig. 3.8: Input H matrix.

i. *Ensure that your files will be stored in the selected folder*

```
Menu: File, Save as, [C:\SIMULIA\User\Ex_3.1\Ex_3.1.cae]
Menu: File, Set Work Directory, [C:\SIMULIA\User\Ex_3.1]
```

ii. *Creating parts*
Due to the symmetry of the plate, only 1/4 of it will be modeled.

```
Module: Part
Menu: Part, Create,
    [Part-1], 3D, Deformable, Shell, Planar, Approx. size [4000], Cont
Menu: Add, Line, Rectangle,
    # Enter the points [0,0], [1000,1000], X # to end the command, Done
```

iii. *Creating materials and sections, and assigning sections to parts*
In Abaqus the matrices A, B, D, H are entered as properties of the section. Notice that, since matrices A, B, D, H contain all the information relevant to the elastic behavior of the laminate, it is not necessary to define a material.

The matrices A, B, D, H are calculated using (3.9), which are implemented in [12]. For the $[0/90]_1$ *laminate, the A, B, D, H matrices are:*

$$[A] = \begin{bmatrix} 711541 & 23317 & 0 \\ \cdot & 711541 & 0 \\ \cdot & \cdot & 43060 \end{bmatrix} [MPa * mm]$$

$$[B] = \begin{bmatrix} -1585193 & 0 & 0 \\ \cdot & 1585193 & 0 \\ \cdot & \cdot & 0 \end{bmatrix} \left[MPa * mm^2\right]$$

$$[D] = \begin{bmatrix} 5929510 & 194306 & 0 \\ \cdot & 5929510 & 0 \\ \cdot & \cdot & 358833 \end{bmatrix} \left[MPa * mm^3\right]$$

$$[H] = \begin{bmatrix} 29442 & 0 \\ \cdot & 29442 \end{bmatrix} [MPa * mm]$$

These are entered into Abaqus/*CAE as follows:*

```
Module: Property
Menu: Section, Create, [Section-1],
    Shell, General Shell Stiffness, Cont
    # In the pop-up window fill the stiffness matrix as follows
    Tab: Stiffness,
    # Upper part of matrix A in rows 1 to 3, columns 1 to 3
    # matrix B in rows 1 to 3, columns 4 to 6
    # Upper part of matrix D in rows 4 to 6, columns 4 to 6
    # The Stiffness matrix should be
    # 711541    23317        0   -1585193         0        0
    #          711541        0          0   1585193        0
    #                    43060          0         0        0
    #                             5929510    194306        0
    #                                       5929510        0
    #                                                 358833
    Tab: Advanced,
    # The values corresponding to the H matrix are entered here
    Transverse Shear Stiffness, # checkmark: Specify values,
    K11: [29442], K12: [0], K22: [29442], OK

Menu: Assign, Section, # pick the part, Done, OK
```

iv. *Creating the Assembly*

```
Module: Assembly
Menu: Instance, Create, Independent, OK
```

v. *Creating steps*

```
Module: Step
Menu: Step, Create, Name [Step-1], Static/General, Cont, OK
```

vi. *Adding BC and loads*

```
Module: Load
Menu: BC, Manager,
    Create, Name [X-symm], Step: Initial,
    Mechanical, Symm/Anti/Enca, Cont
    # pick the left vertical line, Done, XSYMM, OK
    Create, Name [Y-symm], Step: Initial,
    Mechanical, Symm/Anti/Enca, Cont
    # pick the upper horizontal line, Done, YSYMM, OK
    Create, Name [Simple-Support],
    Step: Initial, Mechanical, Disp/Rota, Cont
    # pick right vertical and lower horizontal lines, Done,
    # checkmark: U3, OK, # close BC Manager

Menu: Load, Create, Step: Step-1, Mechanical, Shell edge load, Cont
    # pick the right vertical line, Done, Magnitude [1], OK
```

vii. *Creating mesh*

```
Module: Mesh
Menu: Seed, Instance, Approximate global size [50], Apply, OK
Menu: Mesh, Controls, Element Shape [Quad], Technique [Structured], OK
Menu: Mesh, Instance, # At the bottom of the WS select [Yes]
```

viii. *Solving and visualizing the solution*

```
Module: Job
Menu: Job, Manager
    Create, Name [0-90_1], Cont, OK
    # Job name should be changed for each laminate so the user can
    # access results of different models without re-running
    Data Check # To check the model for errors,
    Submit # To run the model,
    Results # To visualize the solution

Toolbar:    Views, Apply Front View
Menu: Plot, Contours, On Deformed Shape,
    Results, Field Output,
    Output Variable [U], Component [U3], Apply, OK
    # Note that maximum deflection should be -2.196e-01
    # at the upper left corner of the plate
```

ix. *Saving the model database*

```
Menu: File, Save,
Menu: File, Save As [Ex_3.1(1).cae], OK
Menu: File, New Model Database, With Standard/Explicit Model
```

x. *The "Save As" instruction allows the user to keep a copy of the model for the [0/90]*$_1$ *laminate, while the original model is modified to complete the analysis. Also notice that the results of the model can be accessed using*

```
Menu: File, Open, # Browse to C:\SIMULIA\User\Ex_3.1
    File Filter, Output Database (*.odb*), # Select [0-90_1.odb], OK
```

The solution for the remaining laminates with $n > 1$ is similar to the procedure described above. Modifying the number of layers used, without changing the thickness of the laminate, affects only the bending-extension coupling matrix, B. For the additional stacking sequences, the matrix B is found by using [12], which yields

For [0/90]$_5$:

$$[B] = \begin{bmatrix} -317039 & 0 & 0 \\ \cdot & 317039 & 0 \\ \cdot & \cdot & 0 \end{bmatrix} \left[MPa * mm^2\right]$$

For [0/90]$_{10}$:

$$[B] = \begin{bmatrix} -158519 & 0 & 0 \\ \cdot & 158519 & 0 \\ \cdot & \cdot & 0 \end{bmatrix} \left[MPa * mm^2\right]$$

For [0/90]$_{15}$:

$$[B] = \begin{bmatrix} -105468 & 0 & 0 \\ \cdot & 105468 & 0 \\ \cdot & \cdot & 0 \end{bmatrix} \left[MPa * mm^2\right]$$

For [0/90]$_{20}$:

$$[B] = \begin{bmatrix} -79260 & 0 & 0 \\ \cdot & 79260 & 0 \\ \cdot & \cdot & 0 \end{bmatrix} \left[MPa * mm^2\right]$$

Next, modify the model previously saved as Ex_3.1.cae *to use each of the bending-extension coupling matrices for $n > 1$.*

i. *Edit the section property, submit, and visualize*

```
Menu: File, Open, # Browse to ...\Ex_3.1, [Ex_3.1.cae], OK

Module: Property
Menu: Section, Edit, Section-1
    # In the pop-up window modify the cells of the new matrix B
    # rows 1 to 3, columns 4 to 6. The new B sub-matrix is
    # -317039         0    0
    #       0    317039    0
    #       0         0    0
    OK

Module: Job
Menu: Job, Manager, Create, Name [0-90_5], Cont, OK
    Data Check # To check the model for errors,
    Submit # To run the model,
    Results # To visualize the solution

Toolbar:    Views, Apply Front View
Menu: Plot, Contours, On Deformed Shape,
    Results, Field Output,
    Output Variable [U], Component [U3], Apply, OK
```

The maximum deflection should be -2.109×10^{-2} *mm at the upper-left corner of the plate.*

ii. *Save the model database*

```
Menu: File, Save,
Menu: File, Save As [Ex_3.1(5).cae], OK
Menu: File, New Model Database, With Standard/Explicit Model
```

The solution is tabulated in Table 3.2. Bending extension coupling produces a lateral deflection, which diminishes for increasing number of layers n. Note how the bending-extension coupling coefficient $-B_{11}$ *decreases proportionally to the number of layers.*

Table 3.2: Lateral deflection vs. number of layers in Example 3.1.

n	δ $[mm]$	$-B_{11}$	%
1	0.2196	1,585,193	100
5	0.0211	317,039	1/5
10	0.0104	158,518	1/10
15	0.0069	105,468	1/15
20	0.0052	79,260	1/20

3.2.10 Equivalent Orthotropic Input for Laminate FEA

Some FEA codes do not have laminated elements and do not accept the A, B, D, and H matrices as explained in Section 3.2.9. However, if they have orthotropic elements, it is still possible to perform deformation, vibration, and buckling analysis for laminated composites, as it is shown in this section.

Unidirectional Laminate FEA

To model a unidirectional laminate, standard shell elements can be used, even if they are not layered shell, and still obtain correct results of displacements, strains, and stress. The geometry of shells is a surface that represents the midsurface of the real shell. The midsurface is located halfway through the thickness. The positive thickness coordinate points along a normal to the shell midsurface (local z-direction which coincides with the 3-direction). This is the normal definition of shells and it is used in shell elements, as shown in Example 3.2.

Example 3.2 *Use* Abaqus *to model a simply supported rectangular plate with dimensions* $a_x = 4,000$ *mm*, $a_y = 2,000$ *mm, and plate thickness* $t = 10$ *mm. Apply a uniform transverse load* $q_0 = 0.12 \times 10^{-3}$ *MPa. The material is a unidirectional AS4D/9310 lamina (Table 3.1), with the fibers oriented in the x-direction. Determine the deflection of the center point of the plate. Use (1.102–1.104) to calculate the orthotropic stiffness matrix and enter the material as* `Type: Orthotropic`*. This example is continued in Example 3.8, p. 151.*

Solution to Example 3.2 *The thickness coordinate is eliminated from the governing equations so that the 3D problem simplifies to 2D. In the process, the thickness becomes a parameter, which is known and supplied to the modeling software. The model can be constructed and solved as follows:*

```
Menu: File, Save as [C:\SIMULIA\User\Ex_3.2\Ex_3.2.cae]
Menu: File, Set Work Directory [C:\SIMULIA\User\Ex_3.2]
```

i. Creating parts
Due to the symmetry of the plate, only 1/4 of it will be modeled.

```
Module: Part
Menu: Part, Create,
    [Part-1], 3D, Deformable, Shell, Planar, Approx. size [4000], Cont
Menu: Add, Line, Rectangle,
    # Enter the points [0,0], [2000,1000], X # ends the command, Done
```

ii. Creating materials and sections, and assigning sections to parts

```
Module: Property
Menu: Material, Create,
    Mechanical, Elasticity, Elastic, Type: Orthotropic,
    # Enter the values of the components of the stiffness matrix, OK
```

Notice that the stiffness matrix in Abaqus *is built in a different order than the one shown in (1.68). In* Abaqus

$$\left\{\begin{array}{c}\sigma_1\\ \sigma_2\\ \sigma_1\\ \sigma_6\\ \sigma_5\\ \sigma_4\end{array}\right\}=\left[\begin{array}{cccccc}C_{11} & C_{12} & C_{13} & 0 & 0 & 0\\ \cdot & C_{22} & C_{23} & 0 & 0 & 0\\ \cdot & \cdot & C_{33} & 0 & 0 & 0\\ \cdot & \cdot & \cdot & C_{66} & 0 & 0\\ \cdot & \cdot & \cdot & \cdot & C_{55} & 0\\ \cdot & \cdot & \cdot & \cdot & \cdot & C_{44}\end{array}\right]\left\{\begin{array}{c}\epsilon_1\\ \epsilon_2\\ \epsilon_1\\ \gamma_6\\ \gamma_5\\ \gamma_4\end{array}\right\}$$

and $D_{1111}=C_{11}$, $D_{1122}=C_{12}$, $D_{2222}=C_{22}$, $D_{1133}=C_{13}$, $D_{2233}=C_{23}$, $D_{3333}=C_{33}$, $D_{1212}=C_{66}$, $D_{1313}=C_{55}$, $D_{2323}=C_{44}$. *Using the mechanical properties of AS4D/9310 (Table 3.1) to calculate the compliance matrix (1.104), and inverting it, the stiffness matrix is obtained*

$$\left[\begin{array}{cccccc}136222.40 & 3908.70 & 3908.70 & 0 & 0 & 0\\ \cdot & 9251.35 & 3731.31 & 0 & 0 & 0\\ \cdot & \cdot & 9251.35 & 0 & 0 & 0\\ \cdot & \cdot & \cdot & 4306.00 & 0 & 0\\ \cdot & \cdot & \cdot & \cdot & 4306.00 & 0\\ \cdot & \cdot & \cdot & \cdot & \cdot & 2760.00\end{array}\right]\;[MPa]$$

```
Menu: Section, Create,
    Shell, Homogeneous, Cont
    Shell thickness [10], Material [Material-1], OK

Menu: Assign, Section, # pick the part, Done, OK
```

Another option would be to enter the material properties as `Engineering Constants` *(9 values) or* `Lamina` *(5 values*[1]*) in the* `Edit Materials` *window of* Abaqus/*CAE.*

[1] Considering that $G_{12}=G_{13}$ for a transversely isotropic material.

iii. *Creating the assembly*

```
Module: Assembly
Menu: Instance, Create, Independent, OK
```

iv. *Creating analysis steps*

```
Module: Step
Menu: Step, Create, Name [Step-1], Static/General, Cont, OK
```

v. *Adding BC and loads*

```
Module: Load
Menu: BC, Manager,
    Create, Name [X-symm], Step: Initial,
    Mechanical, Symm/Anti/Enca, Cont
    # pick the left vertical line, Done, XSYMM, OK
    Create, Name [Y-symm], Step: Initial,
    Mechanical, Symm/Anti/Enca, Cont
    # pick the upper horizontal line, Done, YSYMM, OK
    Create, Name [Simple-supp], Step: Initial,
    Mechanical, Disp/Rota, Cont
    # pick the right vertical and the lower horizontal lines, Done,
    # checkmark: U3, OK, # close BC Manager

Menu: Load, Create
    Step: Step-1, Mechanical, Pressure, Cont
    # pick the part, Done,
    # Select side for the shell, Brown, Magnitude [0.12e-3], OK
```

vi. *Creating the mesh*

```
Module: Mesh
Menu: Seed, Instance, Approximate global size [50], Apply, OK
Menu: Mesh, Controls, Element Shape [Quad], Technique [Structured], OK
Menu: Mesh, Instance, # At the bottom of the WS select [Yes]
```

vii. *Solving and visualizing the solution*

```
Module: Job
Menu: Job, Manager
    Create, Cont, OK
    Data Check # To check the model for errors,
    Submit # To run the model
    Results # To visualize the solution

Toolbar: Views, Apply Front View
Menu: Plot, Contours, On Deformed Shape,
    Results, Field Output,
    Output Variable [U], Component [U3], Apply, OK
```

The maximum deflection is 17.43 *mm at the center of the plate.*

Symmetric Laminate FEA

If a multidirectional laminate is balanced and symmetric, the apparent laminate orthotropic properties can be found as explained in Section 1.15. The apparent laminate properties represent the stiffness of an equivalent (fictitious) orthotropic plate that behaves like the actual laminate under in-plane loads. These apparent properties should not be used to predict bending response. When the only important response is bending, e.g., a thick cantilever plate under bending, the formulation shown in [1, (6.36)] should be used to obtain the apparent laminate properties. However, in most of the structural design using composite shell, the laminate works under in-plane loads and the formulation in Section 1.15 should be used.

If the laminate is symmetric but not balanced, the axes of orthotropy are rotated with respect to the laminate coordinate system, but still the laminate is equivalent to an orthotropic material as per Section 1.15. For example, a unidirectional laminate oriented at an angle θ with respect to global axes should be modeled on a coordinate system oriented along the fiber direction (see Section 3.2.14).

Example 3.3 *Use* Abaqus *to model a simply supported rectangular plate with dimensions* $a_x = 2000\ mm$, $a_y = 2000\ mm$, *for a laminate* $[\pm 45/0]_S$. *Apply a tensile edge load* $N_x = 200\ N/mm$. *Determine the maximum horizontal displacement. Each layer is* 1.0 *mm thick with the following properties:*

$$\begin{array}{lll} E_1 = 37.88\ GPa & G_{12} = 3.405\ GPa & \nu_{12} = 0.299 \\ E_2 = 9.407\ GPa & G_{23} = 3.308\ GPa & \nu_{23} = 0.422 \end{array}$$

Solution to Example 3.3 *Since the laminate is balanced symmetric, compute the averaged laminate properties* E_x, E_y *and so on using Section 1.15 to yield*

$$\begin{array}{lll} E_x = 20.104\ GPa & G_{xy} = 8.237\ GPa & \nu_{xy} = 0.532 \\ E_y = 12.042\ GPa & G_{yz} = 3.373\ GPa & \nu_{yz} = 0.203 \\ E_z = 10.165\ GPa & G_{xz} = 3.340\ GPa & \nu_{xz} = 0.307 \end{array}$$

The model is implemented in Abaqus *following the steps:*

```
Menu: File, Save as, [C:\SIMULIA\User\Ex_3.3\Ex_3.3.cae]
Menu: File, Set Work Directory, [C:\SIMULIA\User\Ex_3.3]
```

i. Creating parts
Due to the assumed symmetry of the plate, only 1/4 of it will be modeled.

```
Module: Part
Menu: Part, Create,
    [Part-1], 3D, Deformable, Shell, Planar, Approx. size [4000], Cont
Menu: Add, Line, Rectangle,
    # Enter the points [0,0], [1000,1000], X # ends the command, Done
```

ii. Creating materials and sections, and assigning sections to parts

```
Module: Property
Menu: Material, Create,
    Mechanical, Elasticity, Elastic, Type [Orthotropic],
# Enter the values of the components of the stiffness matrix, OK
```

Notice that the stiffness matrix in Abaqus *is built in a different order than the one shown in (1.68). In* Abaqus

$$\left\{\begin{array}{c}\sigma_1\\\sigma_2\\\sigma_1\\\sigma_6\\\sigma_5\\\sigma_4\end{array}\right\}=\left[\begin{array}{cccccc}C_{11}&C_{12}&C_{13}&0&0&0\\\cdot&C_{22}&C_{23}&0&0&0\\\cdot&\cdot&C_{33}&0&0&0\\\cdot&\cdot&\cdot&C_{66}&0&0\\\cdot&\cdot&\cdot&\cdot&C_{55}&0\\\cdot&\cdot&\cdot&\cdot&\cdot&C_{44}\end{array}\right]\left\{\begin{array}{c}\epsilon_1\\\epsilon_2\\\epsilon_1\\\gamma_6\\\gamma_5\\\gamma_4\end{array}\right\}$$

and $D_{1111}=C_{11}$, $D_{1122}=C_{12}$, $D_{2222}=C_{22}$, $D_{1133}=C_{13}$, $D_{2233}=C_{23}$, $D_{3333}=C_{33}$, $D_{1212}=C_{66}$, $D_{1313}=C_{55}$, $D_{2323}=C_{44}$. *Using the mechanical properties given above to calculate the compliance matrix (1.104), and inverting it, the stiffness matrix is*

$$\left[\begin{array}{cccccc}26677.59&9531.84&5736.79&0&0&0\\\cdot&15897.74&4216.73&0&0&0\\\cdot&\cdot&11813.63&0&0&0\\\cdot&\cdot&\cdot&8264.46&0&0\\\cdot&\cdot&\cdot&\cdot&3378.38&0\\\cdot&\cdot&\cdot&\cdot&\cdot&3344.48\end{array}\right][MPa]$$

which is entered into Abaqus*/CAE as follows:*

```
Menu: Section, Create,
    Shell, Homogeneous, Cont
    Shell thickness [6], Material [Material-1], OK

Menu: Assign, Section, # pick the part, Done, OK
```

Alternatively, one could enter the material properties as ***`Engineering Constants`*** *(9 values) in the* ***`Edit Materials`*** *window of* Abaqus*/CAE.*

iii. *Creating the assembly*

```
Module: Assembly
Menu: Instance, Create, Independent, OK
```

iv. *Creating analysis steps*

```
Module: Step
Menu: Step, Create, Name [Step-1], Static/General, Cont, OK
```

v. *Adding BC and loads*

```
Module: Load
Menu: BC, Manager,
    Create, Name [X-symm], Step: Initial, Mechanical,
    Symm/Anti/Enca, Cont
    # pick the left vertical line, Done, XSYMM, OK
    Create, Name [Y-symm], Step: Initial, Mechanical,
    Symm/Anti/Enca, Cont
    # pick the upper horizontal line, Done, YSYMM, OK
    Create, Name [Simple-supp], Step: Initial, Mechanical,
    Disp/Rota, Cont
    # pick the right vertical and the lower horizontal lines, Done,
    # checkmark: U3, OK, # close BC Manager

Menu: Load, Create
Step: Step-1, Mechanical, Shell edge load, Cont
    # pick the right vertical line, Done, Magnitude [-200], OK
```

vi. Creating the mesh

```
Module: Mesh
Menu: Seed, Instance, Approximate global size [50], Apply, OK
Menu: Mesh, Controls, Element Shape [Quad], Technique [Structured], OK
Menu: Mesh, Instance, # At the bottom of the WS select [Yes]
```

vii. Solving and visualizing the solution

```
Module: Job
Menu: Job, Manager
    Create, Cont, OK
    Data Check # To check the model for errors,
    Submit # To run the model
    Results # To visualize the solution

# ‘Apply Front View’ from the ‘Views toolbar’,
Menu: Plot, Contours, On Deformed Shape,
    Results, Field Output,
    Output Variable [U], Component [U1], Apply, OK
```

The maximum displacement should be 1.667 *at the right edge of the plate.*

The resulting maximum horizontal displacement on a quarter-plate model is 1.667 *mm. The planes $x = 0$ and $y = 0$ are not symmetry planes for a* $[\pm 45/0]_S$ *but once the laminate is represented by equivalent orthotropic properties, as is done in this example, the lack of symmetry at the lamina level is lost and it does not have any effect on the mid-surface displacements. Therefore, one-quarter of the plate represents well the entire plate as long as no stress analysis is performed. Furthermore, displacement and mid-surface strain analysis can be done with the laminate replaced by an equivalent orthotropic material. However, even if the full plate were to be modeled, the stress values in the equivalent orthotropic material are not the actual stress values of the laminate. While the material analyzed in this example is not homogeneous, but laminated, the material in Example 3.2 is a homogeneous unidirectional material. Therefore, the stress values are not correct in this example but they are correct in Example 3.2, p. 113.*

Asymmetric Laminate FEA

If the laminate is not symmetric, bending-extension coupling must be considered. Strictly speaking, such material is not orthotropic and should not be modeled with an equivalent laminate material. Even then, if only orthotropic shell elements are available and the bending-extension coupling effects are not severe, the material could be approximated by an orthotropic material by neglecting the matrices B and D. The ratios defined in [1, (6.37–6.39)] can be used to assess the quality of the approximation obtained using apparent elastic properties. Care must be taken for unbalanced laminates that the A and H matrix are formulated in a coordinate system coinciding with the axes of orthotropy of the laminate.

3.2.11 LSS for Multidirectional Laminate FEA

For computation of strain and stress at the mesoscale (lamina level), it is necessary to know the description of the laminate and the properties of each lamina. The description of the multidirectional laminate includes the LSS, which specifies the angle of each lamina with respect to the x-axis of the laminate, the thickness, and the elastic material properties of each lamina. Then, the software computes the matrices A, B, D, and H internally. In this way, the software can compute the stress components in each layer. This approach is illustrated in the Example 3.4, p. 121. In *Abaqus*/CAE, the LSS can be specified in two ways, using Composite Sections or Composite Layups.

Composite Section

Composite Section is the basic tool that allows the user to define the LSS. It does not differ from the classical definition and assignment of all other types of sections such as *solid, shell, beam, truss,* and so on. First, one defines one or more materials, which in this case could be type *isotropic, engineering constants, lamina, orthotropic,* or *anisotropic.* In CAE, this is done as follows:

```
Module: Property
Menu:   Material, Create, Mechanical, Elasticity, Elastic, Type: Lamina
```

Second, one defines a laminated section as follows.

```
Module: Property
Menu:   Section, Create, Category: Solid, Composite, Cont
    Material: [select], Thickness: [value], Orientation Angle: [value],
    Integration Points: [value], Ply name: [value]
```

Note that *category* can be *shell* or *solid* depending on what type of element you intend to use. *Continuum shell elements* (e.g., SC8R), *thin shell elements* (e.g., S8R5), and *general-purpose shell elements* (e.g., S8R) need a *shell, composite* section. Solid elements, such as C3D20, need a *solid, composite* section.

Solid elements can be laminated but the maximum order of variation of the displacements is quadratic. Therefore, the strain variation can be at most linear.

This is often insufficient to model the variation of strain through the thickness of a laminate. To alleviate this problem, solid elements can be stacked, even at one element per lamina. However, due to aspect ratio limitations a solid element with the thickness of a lamina should not have the other two dimensions larger than about 10 times the thickness. This would lead to a very refined mesh that would result in a computationally expensive solution.

For each section defined, the *Edit Section* window allows the user to define the LSS by giving the following data for each lamina in the laminate: *Material, Thickness, Orientation Angle, Integration Points,* and *Ply Name.* Ply Name is optional, but recommended because it helps understand the results. The material is chosen from the materials previously defined. Thickness and orientation are required input data. For integration points, it is best to use the default, which is 3 points using the Simpson rule. In this way, one point is on each surface and one in the middle of each lamina. This is optimal for results visualization provided one remembers to modify the *field output requests* as follows:

```
Module: Step
Menu:   Output, Field Output Requests, Edit, F-Output-1,
        Output at shell, Specify: 1,2,3
```

where $1, 2, 3, ..., 3N$, are the Simpson points through the thickness of the shell, where N is the number of laminas in the laminate. Then, visualization is done by selecting

```
Module: Visualization
Menu:   Results, Section Points, Selection method: Plies,
        Plies: [select],
        Ply result location: [Bottommost, Middle/Single, Topmost,
            Top & Bottom]
```

If one selects Top & Bottom, the results on the faces of the laminate are shown. If a face is not in view, just rotate the model to see it.

Third, one must *assign* sections to *regions* in the classical way. If using *conventional shell elements* and *general-purpose shell elements*, the *regions* will be a flat or curved surfaces, i.e., 2D cells. If using *continuum shell elements*, the regions will be 3D cells.

If using *conventional shell elements* and *general-purpose shell elements*, the entire LSS is defined in a *composite section* and assigned to a *region. Continuum shell elements* may span the entire thickness of the shell, in which case the assignment is similar, with one composite section assigned to each flat/curved surface. However, continuum shell elements can be stacked. This is accomplished by partitioning the thickness of the 3D cell to delimit the regions for each continuum shell element. In this case, a different composite section can be assigned to each region (cell). Each continuum shell element will be laminated as mandated by its associated composite section definition.

Composite Layup

Composite Layup is an alternate way of defining the LSS. The concept of region is different than the usual concept of region in classical CAE. A *composite layup* must be associated to a single element through the thickness of a region. If you refine the mesh through the thickness of a region (cell) spanned by a composite layup, each element though the thickness will be laminated as specified by the composite layup, which would lead to erroneous results.

With composite layup, the user does not create a section nor assign a section to a region. Rather, the *composite layup* does the definition and assignment simultaneously. If by mistake, both *composite section* and *composite layup* are used for the same region, *composite layup* overwrites the former.

A composite layup window is invoked as follows:

```
Module: Property
Menu:   Composite, Create
```

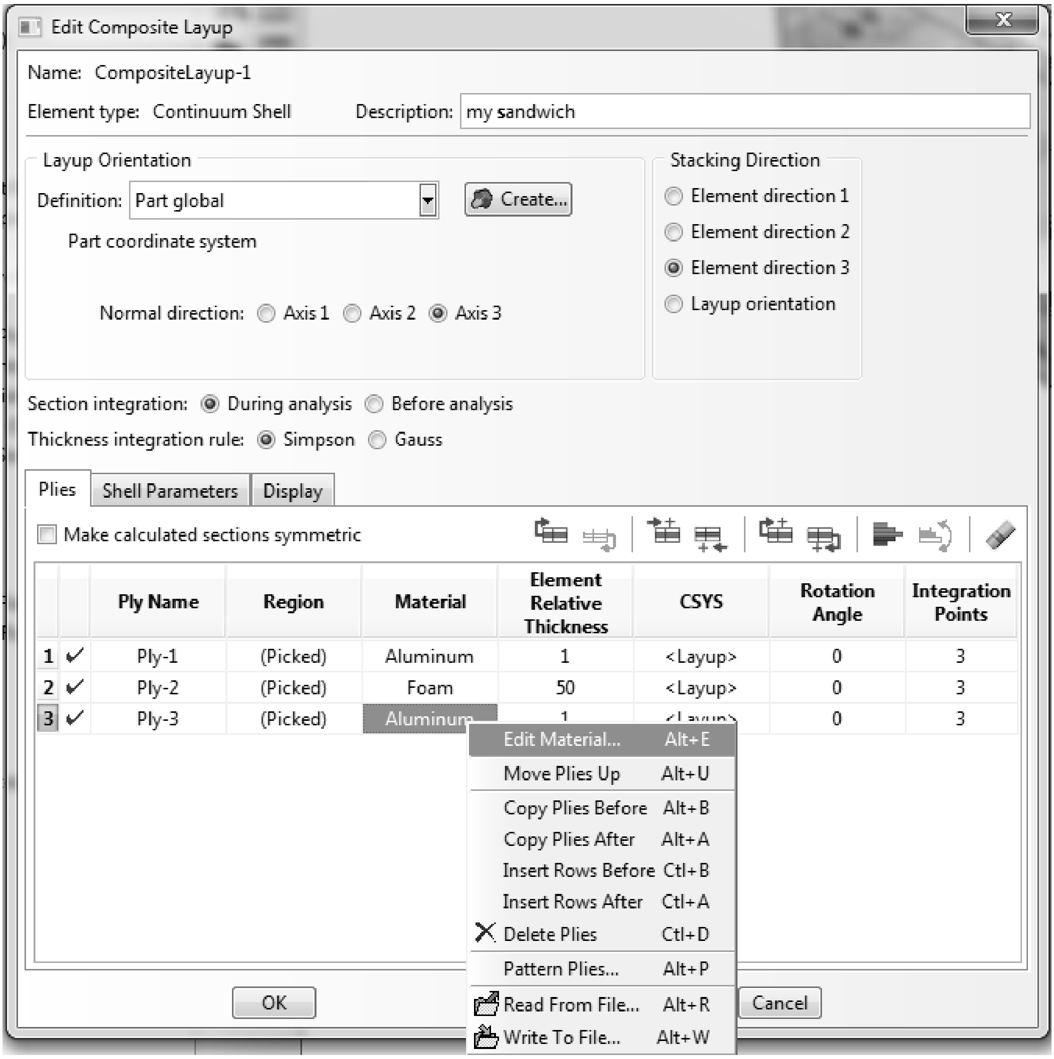
At this point one must decide what type of section to create: Conventional Shell, Continuum Shell, or Solid. *Continuum shell elements* (e.g., SC8R), *thin shell elements* (e.g., S8R5), and *general-purpose shell elements* (e.g., S8R) need a *shell, composite* section. Solid elements, such as C3D20, need a *solid, composite* section.

The Edit Composite Layup window (Figure 3.9) has four tabs: Plies, Offset, Shell Parameters, and Display. The LSS is specified in the Plies tab. The position of the reference surface is given in the Offset tab. The thickness modulus is specified in the Shell Parameters tab.

The Plies tab has seven columns: Ply Name, Region, Material, Relative Thickness, CSYS, Rotation Angle, and Integration Points. Ply Name is optional, but recommended because it helps understand the results. Region ties the ply to a region. The material is chosen from the materials previously defined. The ply relative thickness are proportionality factors used to determine the thickness of the ply as a function of the thickness of the element, which is given by the mesh. They do not need to be actual thicknesses or add up to 1.0. CSYS is used to have additional flexibility in specifying the ply orientation with respect to the global coordinate system of the model. Basically, CSYS allows the user to rotate the laminate coordinate system as a whole. Rotation Angle is the ply angle θ_k with respect to the laminate coordinate system. For integration points, it is best to use the default, which is 3 points using the Simpson rule. In this way, one point is on each surface and one in the middle of each lamina. This is optimal for results visualization provided one remembers to modify the *field output requests* as explained in Section 3.2.11.

Example 3.4 *Consider a simply supported square plate* $a_x = a_y = 2000$ *mm,* $t = 10$ *mm thick, laminated with AS4D/9310 (Table 3.1) in a* $[0/90/\pm 45]_S$ *symmetric laminate configuration. The plate is loaded with a tensile load* $N_x = 100$ *N/mm* ($N_y = N_{xy} = M_x = M_y = M_{xy} = 0$)*. Compute the in-plane shear stress in lamina coordinate system* σ_6 *using four modeling techniques:*

a. with conventional shell elements,



Edit Composite Layup

Name: CompositeLayup-1

Element type: Continuum Shell Description: my sandwich

Layup Orientation

Definition: Part global

Part coordinate system

Normal direction: ○ Axis 1 ○ Axis 2 ● Axis 3

Stacking Direction

○ Element direction 1
○ Element direction 2
● Element direction 3
○ Layup orientation

Section integration: ● During analysis ○ Before analysis

Thickness integration rule: ● Simpson ○ Gauss

□ Make calculated sections symmetric

		Ply Name	Region	Material	Element Relative Thickness	CSYS	Rotation Angle	Integration Points
1	✓	Ply-1	(Picked)	Aluminum	1	<Layup>	0	3
2	✓	Ply-2	(Picked)	Foam	50	<Layup>	0	3
3	✓	Ply-3	(Picked)	Aluminum	1	<Layup>	0	3

Fig. 3.9: Entering the laminate stacking sequence using the `Composite Layup` feature.

b. with general-purpose shell elements,
c. with continuum shell elements in a composite section with a single element through the plate thickness, and
d. with continuum shell elements in a composite layup with several elements through the plate thickness.

This example is continued in Example 3.12, p. 167.

Solution to Example 3.4 *Parts (a) and (b) involve* conventional shell elements *using a mesh without thickness. Therefore, most of the model for parts (a) and (b) is independent of the type of element. Note the LSS is given starting at layer #1 at the bottom of the laminate. The full plate is modeled without taking advantage of symmetry, and nodes are constrained to illustrate how to constrain a model to prevent rigid body motion. The following pseudo code illustrates the whole procedure. If in doubt, open* `Ex_3.4.cae` *available in [5].*

i. Make sure the .cae file is saved in the right place. You can also set the `Work Directory` *to the same folder where you saved the* `.cae` *file*

```
Menu:   File, Save as [c:\simulia\user\Ex_3.4\Ex_3.4.a.cae]
Menu:   File, Set Work Directory [c:\simulia\user\Ex_3.4]
```

ii. Creating parts

```
Module: Part,
Menu:   Part, Create,
Name [Part-1], 3D, Deformable, Shell, Planar, Approx size [2000], Cont
Menu:   Add, Line, Rectangle, [-1000,-1000], [1000,1000], X, Done
```

iii. Creating materials and sections, and assigning sections to parts

```
Module: Property
Menu:   Material, Create, Name [Material-1], Mech., Elast., Elastic,
    Type [Engineering Constants],
    # Note: you can cut and paste an entire row of values from Excel
    [1.3386E+05    7.7060E+03    7.7060E+03    3.0100E-01 3.0100E-01
    3.9600E-01    4.3060E+03    4.3060E+03    2.7600E+03], OK
Menu:   Section, Create, Shell, Composite, Cont
    # checkmark: Symmetric layers
    Integration [Simpson]
    # insert rows with right click until you have 4 rows
    Material [Material-1]   # for all
    Thickness [1.25] #for all
    Orientation [0,90,45,-45]
    Integration points [3,3,3,3] # results at bot/mid/top of each ply
    Ply Names [k1,k2,k3,k4], OK, # k1:bottom, k4:middle
    # due to symmetry, CAE adds [-45,45,90,0] to the model,
Menu:   Assign, Section, # pick part, Done, OK
```

iv. Creating assembly

```
Module: Assembly
Menu:   Instance, Create, Independent, OK
```

v. *Creating analysis steps*

```
Module: Step
Menu:   Step, Create, Name [Step-1], Static/General, Cont, OK
Menu:   Output, Field Output Requests, Edit, F-Output-1
    Step: Step-1, Output variables:
    # unselect everything, # select only what you need: S, E, U, RF
    # request at 8x3=24 points (8 layers, 3 Simpson points per layer)
    # Output at shell and layered section points:
    Specify: [1,...24], OK # list all #'s from 1 to 24
```

vi. *Adding loads*

```
Module: Load
    # since the part is a shell, an edge load is: Shell edge load
    Menu:   Load, Create, Name [Load-1], Step: Step-1, Mechanical,
    Shell edge load, Cont
    # pick the edges x=-1000 and x=1000, Done, Magnitude [-100], OK
```

In order to identify sets of nodes to constrain, when setting the BC, it is required to mesh the model now.

vii. *Creating mesh*

```
Module: Mesh
Menu:   Seed, Instance, Approx global size [100], Apply, OK
Menu:   Mesh, Controls, Element Shape [Quad], Technique [Structured],
    OK
Menu:   Mesh, Element Type, Std, Quad, DOF [5], OK # element S8R5
Menu:   Mesh, Instance, Yes
Menu:   Tools, Set, Manager
    Create, [center], Node, Cont, # center node, Done
    Create, [y-hold], Node, Cont, # nodes at (-1000,0) and (1000,0),
    Done
    Create, [edges],  Node, Cont, # nodes at the edges of the plate,
    Done
    # close Set Manager
```

viii. *Adding BC*

```
Module: Load
Menu:   BC, Manager
    Create, Name [edges], Step: Initial, Mechanical, Disp/Rota, Cont
    Sets, # in the Region Selection pop-up window, [edges], Cont
    # checkmark: U3, OK
    Create, Name [center], Step: Initial, Mechanical, Disp/Rota, Cont
    # in the Region Selection pop-up window, [center], Cont
    # checkmark: U1, U2, and U3, OK
    Create, Name [y-hold], Step: Initial, Mechanical, Disp/Rota, Cont
    # in the Region Selection pop-up window, [y-hold], Cont
    # checkmark: U2, and U3, OK
    # close BC Manager
```

ix. *Solving and visualizing the solution*

```
Module: Job
Menu:   Job, Manager
    Create, Cont, OK,
    Submit  # to run the model
    Results # to visualize the solution

Menu:   Viewport, Viewport Annotation Options, Tab: State Block,
    Set font, Size [12], Apply to [# checkmark all], OK, OK
Menu:   Plot, Undeformed Shape,
Menu:   Plot, Allow Multiple Plot States,
Menu:   Plot, Deformed Shape,
Menu:   Plot, Material Orientations, On Deformed Shape
    # by default shows the 0-deg layer on bottom
Menu:   Result, Section Points, Plies, # select K1, Apply
    # repeat for all the plies to visualize their orientation, OK
Menu:   Plot, Contour, On Deformed Shape,
    Field Output Toolbox, Primary, S, S12 # laminate stress_6
Menu:   Result, Section Points, Plies, # select K1, Apply
    # repeat for all the plies to visualize S12, OK
    # select ply K1, Apply  # sigma_6(K1) =  0.0
    # select ply K2, Apply  # sigma_6(K2) =  0.0
    # select ply K3, Apply  # sigma_6(K3) = -1.112
    # select ply K4, Apply  # sigma_6(K4) =  1.122, OK
```

(a) Solution using thin shell elements

In the procedure given above, selection of a conventional shell element S8R5 is illustrated by:

```
Module: Mesh
Menu:   Mesh, Element Type, Standard, Quadratic, DOF [5], OK
```

which results in element S8R5 being chosen. It is convenient to save the solution of this problem, and modify the model to solve part (b).

```
Menu:   File, Save as [c:\simulia\user\Ex_3.4\Ex_3.4.a.cae], OK
```

Visualization of lamina stresses in lamina coordinate system results in

$$\sigma_6 = 0, 0, -1.112, 1.112\, MPa$$

in the $0, 90, 45^o, -45^o$ plies, respectively. These results can be verified by comparison with the classical lamination theory (CLT) results obtained with [12].

(b) Solution using general-purpose shell elements

To change the analysis from thin shell to general-purpose shell, one only needs to change the element type, by selecting `DOF [6]`*, as follows:*

```
Menu:   File, Save as [c:\simulia\user\Ex_3.4\Ex_3.4.b.cae], OK
Module: Mesh
Menu:   Mesh, Element Type, Standard, Quadratic, DOF [6], OK # S8R
```

This results in element S8R being chosen–a doubly-curved, general-purpose (thick/thin) shell with reduced integration. The job must be resubmitted for execution. Since this example does not involve bending, the results are identical to those of part (a).

(c) Solution using continuum shell elements in a composite shell

To use continuum shell elements, the geometric model must be that of a solid, in this case with dimensions $2,000 \times 2,000 \times 10$ *mm.*

i. *Saving the model, and setting the* `Work Directory`

```
Menu:   File, Save as [c:\simulia\user\Ex_3.4\Ex_3.4.c.cae]
Menu:   File, Set Work Directory [c:\simulia\user\Ex_3.4]
```

ii. *Creating parts*

```
Module: Part
Menu:   Part, Create,
    Name [Part-1], 3D, Deformable, Solid, Extrusion, size [2000], Cont
Menu:   Add, Line, Rectangle, [-1000,-1000], [1000,1000], X, Done,
    Depth [10], OK
```

iii. *Creating materials and sections, and assigning sections to parts*

```
Module: Property
Menu: Material, Create, Name [Material-1], Mech., Elast., Elastic,
    Type [Engineering Constants],
    # Note: you can cut and paste an entire row of values from Excel
    [1.3386E+05    7.7060E+03    7.7060E+03    3.0100E-01 3.0100E-01
    3.9600E-01    4.3060E+03    4.3060E+03    2.7600E+03], OK
Menu:   Section, Create, [Section-1], Shell, Composite, Cont
    # insert rows with right click until you have 8 rows
    Material [Material-1]   # for all, Thickness [1.25] # for all
    Orientation [0,90,45,-45,-45,45,90,0],
    Integration points [3] # for all
    Ply Names [PLY-1,PLY-2,PLY-3,PLY-4,PLY-5,PLY-6,PLY-7,PLY-8], OK
Menu:   Assign, Section, # pick the part, Done, OK
```

iv. *Creating assembly*

```
Module: Assembly
Menu:   Instance, Create, Independent, OK
```

v. *Creating analysis steps*

```
Module: Step
Menu:   Step, Create, Name [Step-1], Static/General, Cont, OK
Menu:   Output, Field Output Requests, Edit, F-Output-1
    Step:   Step-1, Output variables:
    # unselect everything, # select only what you need: S, E, U, RF
    # request at 8x3=24 points (8 layers, 3 Simpson points per layer)
    # Output at shell and layered section points:
    Specify: [1,...24], OK # 1,2,3,4,...
```

vi. *Adding loads*

```
Module: Load
Menu:   Load, Create
    Name [Load-1], Step: Step-1, Mechanical, Surface traction, Cont
    # an edge load of 100 N/mm on a surface 10 mm-thick equals 10 N/mm2
    # Zoom into the upper-right corner of the plate
    # pick the surface x=1000, Done,
    Distribution [Uniform], Traction [General],
    Vector, Edit, [0,0,0], [1,0,0], Magnitude [10], OK
    # repeat the procedure to apply a surface traction of 10 N/mm2
    # on the x=-1000 surface, # Name [Load-2] and direction [-1,0,0]
```

In order to identify the nodes to constrain, when setting the BC, it is required to mesh the model now.

vii. *Creating mesh*

```
Module: Mesh
Menu:   Seed, Instance, Approx global size [100], OK
Menu:   Mesh, Controls, # if required select the whole plate, Done,
    Element Shape [Hex], Technique [Sweep], OK
Menu:   Mesh, Element Type, # if required select whole plate, Done,
    Family [Continuum Shell], # element SC8R has been selected, OK
Menu:   Mesh, Instance, Yes
    # create sets of nodes to facilitate applying the BC to the model
    Views Toolbar, Front View # to visualize better
Menu:   Tools, Set, Manager
    Create, Name [edges], Type [Node], Cont
    # select all the nodes along the edges of the plate, Done
    Create, Name [center], Type [Node], Cont
    # select the nodes at the center of the plate (x=0,y=0), Done
    Create, Name [y-hold], Type [Node], Cont
    # select the nodes at (x=-1000,y=0) and (x=1000,y=0), Done
    # close the Set Manager pop-up window
```

viii. *Applying BC*

```
Module: Load
Menu:   BC, Manager
    Create, Name [edges], Step: Initial, Mechanical, Disp/Rota, Cont
    Sets, # in the Region Selection pop-up window, [edges], Cont
    # checkmark: U3, OK
    Create, Name [center], Step: Initial, Mechanical, Disp/Rota, Cont
    # in the Region Selection pop-up window, [center], Cont
    # checkmark: U1, U2, and U3, OK
    Create, Name [y-hold], Step: Initial, Mechanical, Disp/Rota, Cont
    # in the Region Selection pop-up window, [y-hold], Cont
    # checkmark: U2, and U3, OK
    # close BC Manager
```

ix. *Solving and visualizing the solution*

```
Module: Job
Menu:   Job, Manager
```

```
        Create, Cont, OK,
        Submit  # to run the model
        Results # to visualize the solution

    Menu:   Viewport, Viewport Annotation Options, Tab: State Block,
        Set font, Size [12], Apply to [# checkmark all], OK, OK
    Menu:   Plot, Undeformed Shape,
    Menu:   Plot, Allow Multiple Plot States,
    Menu:   Plot, Deformed Shape,
    Menu:   Plot, Material Orientations, On Deformed Shape
    Menu:   Result, Section Points, Plies, # select PLY-1, Apply
        # repeat for all the plies to visualize their orientation, OK
    Menu:   Plot, Contour, On Deformed Shape,
        Field Output Toolbox, Primary, S, S12 # laminate stress_6
    Menu:   Result, Section Points, Plies, # select PLY-1, Apply
        # repeat for all the plies to visualize S12 stress, OK
```

(d) Solution using continuum shell elements in a composite layup

In order to use a Composite Layup to model the laminate the solid part must be divided in regions, each one corresponding to a layer of the laminate. The creation of the part must be modified as

i. Creating parts

```
    Module: Part
    Menu:   Part, Create,
        Name [Part-1], 3D, Deformable, Solid, Extrusion, size [2000], Cont
    Menu:   Add, Line, Rectangle, [-1000,-1000], [1000,1000], X, Done,
        Depth [10], OK
        # zoom into the upper-right corner of the plate
        # divide the thickness of the plate in 8 segments
    Menu:   Tools, Partition, Type [Edge], Method [Enter parameter]
        # pick edge, Done, Normalized edge param [0.5], Create Partition
        # repeat procedure until the edge is divided in 8 segments, Done
        # in the Create Partition pop-up window select
        Type [Cell], Method [Define cutting plane], Point & Normal,
        # select one point defined previously (start at top of the plate),
        # select edge used as normal direction (along positive z-axis),
        Create Partition # create first region representing the top ply
        # pick region to partition, Done,
        Point & Normal, # repeat procedure,
        # once 8 regions have been created, Done, # close pop-up window
```

The definition of the material properties must be modified as

ii. Creating materials and sections, and assigning sections to parts

```
    Module: Property
    Menu:   Material, Create, Name [Material-1], Mech., Elast., Elastic,
        Type [Engineering Constants],
        # Note: you can cut and paste an entire row of values from Excel
        [1.3386E+05   7.7060E+03   7.7060E+03   3.0100E-01   3.0100E-01
        3.9600E-01    4.3060E+03    4.3060E+03    2.7600E+03], OK
    Menu:   Composite, Create
        Initial ply count [8], Element Type [Continuum Shell], Cont
        Thickness integration rule [Simpson]
        # in column named Region double-click cell corresponding Ply-1,
        # pick top region of laminate, Done, # repeat for other plies
        # in column named Material double-click cell corresponding Ply-1,
        Material [Material-1], OK
        # repeat to assign Material-1 to the other plies
        Element Relative Thickness [1] # for all plies
        Rotation Angle [0, 90, 45, -45, -45, 45, 90, 0] # one per ply
        Integration points [3] # for all plies, OK
```

The rest of the solution follows the same steps described in part (c).

3.2.12 FEA of Ply Drop-Off Laminates

Sometimes it is convenient to set the reference surface at the bottom (or top) of the shell. One such case is when the laminate has ply drop-offs, as shown in Figure 3.10. When the design calls for a reduction of laminate thickness, plies can be gradually terminated from the thick to the thin part of the shell. As a rule of thumb, ply drop-off should be limited to a 1:16 to 1:20 ratio ($Th : L$ ratio in Figure 3.10) unless detailed analysis and/or testing supports a steeper drop-off ratio. For this case, it is convenient to specify the geometry of the smooth surface, or tool surface.

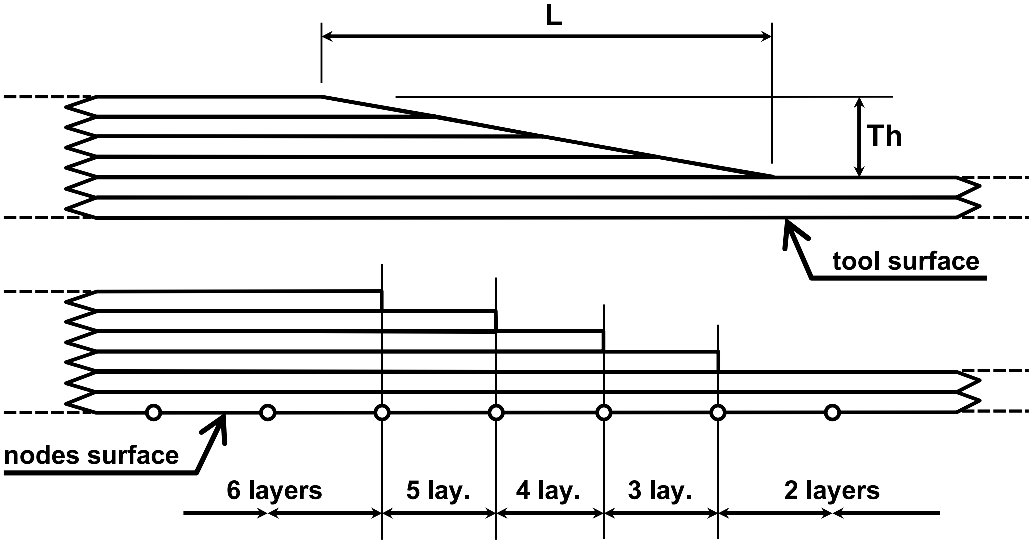


Fig. 3.10: Ply drop-off of length (L) and thickness (Th) and FE model simplifications.

Then, every time a ply or set of plies is dropped, the material and thickness for those elements is changed. This is illustrated in the next examples. Not all software has this capability and it may be necessary to assume that the midsurface is smooth while in reality only the tool surface is smooth. As long as the thickness is small compared to the other two dimensions of the structure, such assumption is unlikely to have a dramatic effect in the results of a global analysis, such as deformation, buckling, and even membrane stress analysis. The exact description of the thickness geometry begins to play a role when detailed 3D stress analysis of the ply drop-off region is required, but at that point, a 3D local model is more adequate.

Example 3.5 *A composite strip* 120 *mm long and* 100 *mm wide has a ply drop-off between laminate A:* $[90/0]_S$*, and laminate B:* $[90/0]$*, with ply thickness* 0.75 *mm. A ply drop-off ratio 1:20 is used, meaning that the transition length is 20 times the dropped thickness. The strip is under tension* $N = 10$ *N/mm applied to the bottom edges on the strip. The material is AS4D/9310 (Table 3.1). Use symmetry to model 1/2 of the tape. Visualize and report the maximum deflection (in the Z-direction, normal to the plate). Also, tabulate the maximum values of* σ_{11} *at the top and bottom surface of each lamina (not the minimum, not the maximum absolute value).*

Solution to Example 3.5 *Three different sections are defined: one for laminate A, one for laminate B, and one to model the ply drop-off between them (called laminate AB), where only the top* 90° *lamina has been dropped. The thickness of the drop-off is equivalent to two laminas,* $0.75 \times 2 = 1.5$ *mm. With a ply drop-off ratio 1:20, the total length of ply drop-off is* $1.5 \times 20 = 30$ *mm. If the laminate A spans* $0 < x < 60$ *mm, the transition goes from* $x = 60$ *mm to* $x = 90$ *mm, and laminate B from* $x = 90$ *mm to the loaded edge at* $x = 120$ *mm.*

The midsurface of each laminate (A, B, and AB) is offset at different z-distances with respect to the flat bottom surface. Therefore, it is preferred to apply the load at the common bottom surface. This however introduces a bending moment because the traction load is offset with respect to the midsurface of the loading edge. This effect can be easily corrected by applying a bending moment equal to the product of the traction times one half the thickness of the laminate at the loading edge, but it has not been done in this example to emphasize the resulting deformation.

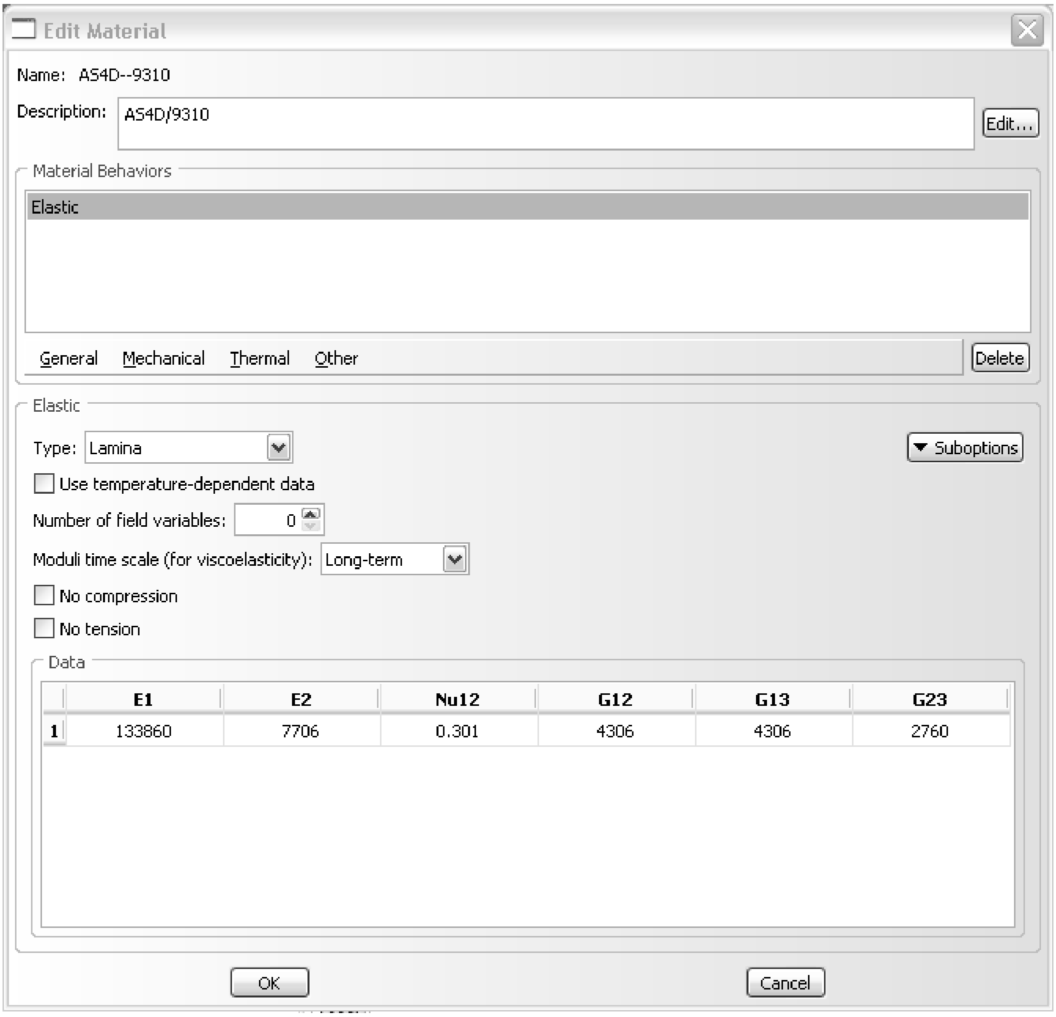
The bottom lamina is designated as #1. Additional laminas are stacked from bottom to top in the positive normal z-direction of the element coordinate system. With the surface of the plate resting on the x-y plane, the x-z plane is a plane of symmetry (YSYMM). Then, the thicker end of the strip (laminate A) on the left of the model can be properly supported by imposing an `Encastre-condition`*; this will guarantee symmetry conditions with respect to the y-z plane.*

The material properties are defined in the `Edit Material` *window (Figure 3.11), but this is quite tedious. If the properties are listed in* `Excel`*, or in a Tab-separated file, one can cut and paste directly into the* `Edit Material` *window. Otherwise, they can be accessed directly from the CADEC database using a Web service [18].*

The laminate stacking sequence (LSS) is defined in the `Edit Section` *window, which is reached as follows:*

```
Module: Property
Menu:   Section, Create, Shell, Composite, Cont
```

In the `Edit Section` *window (Figure 3.12), the first row corresponds to lamina #1, which is located at the bottom of the laminate. Subsequent rows are used to define LSS of a laminate with n laminas. The last row corresponds to the topmost lamina in the laminate. Note how we have assigned the names k1, k2, k3, k4, to the laminas, following the usual notation where lamina #1 is at the bottom of the laminate.*



Name: AS4D--9310

Description: AS4D/9310

Material Behaviors

Elastic

Elastic

Type: Lamina

□ Use temperature-dependent data

Number of field variables: 0

Moduli time scale (for viscoelasticity): Long-term

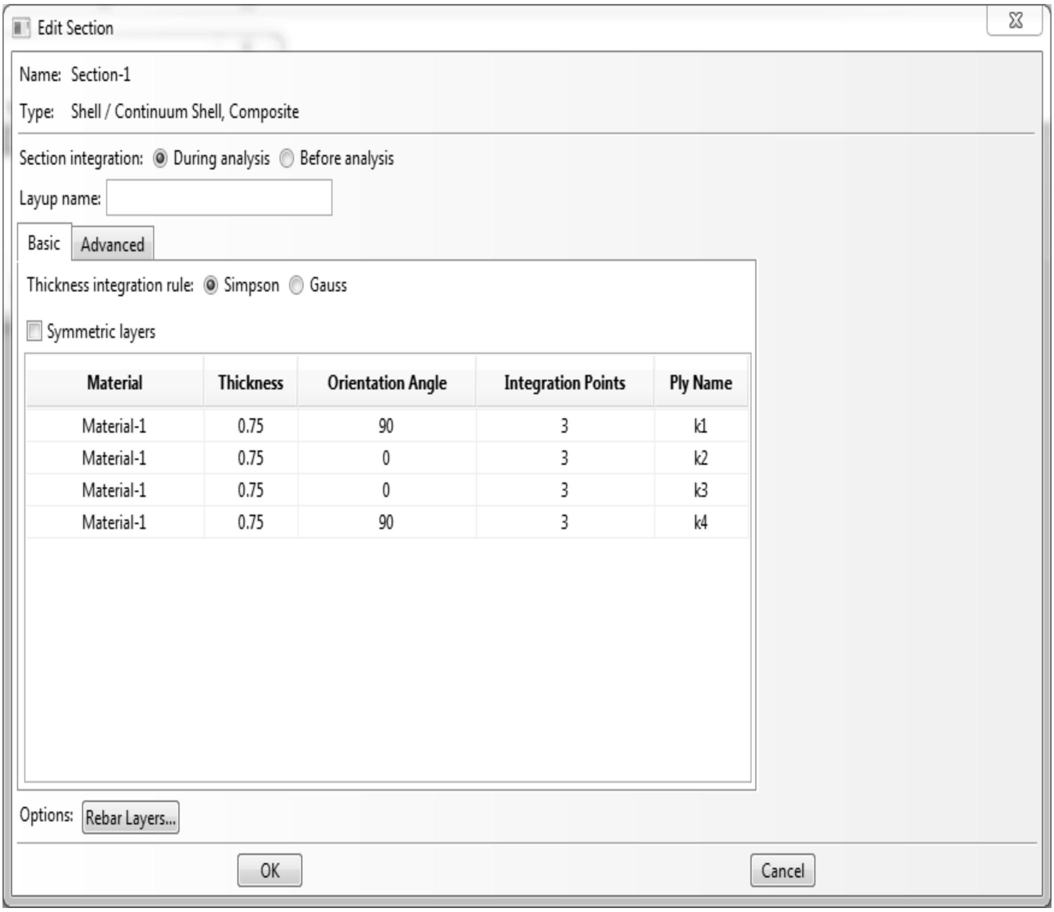
□ No compression

□ No tension

Data

	E1	E2	Nu12	G12	G13	G23
1	133860	7706	0.301	4306	4306	2760

Fig. 3.11: The Edit Material window.



Edit Section

Name: Section-1

Type: Shell / Continuum Shell, Composite

Section integration: ● During analysis ○ Before analysis

Layup name:

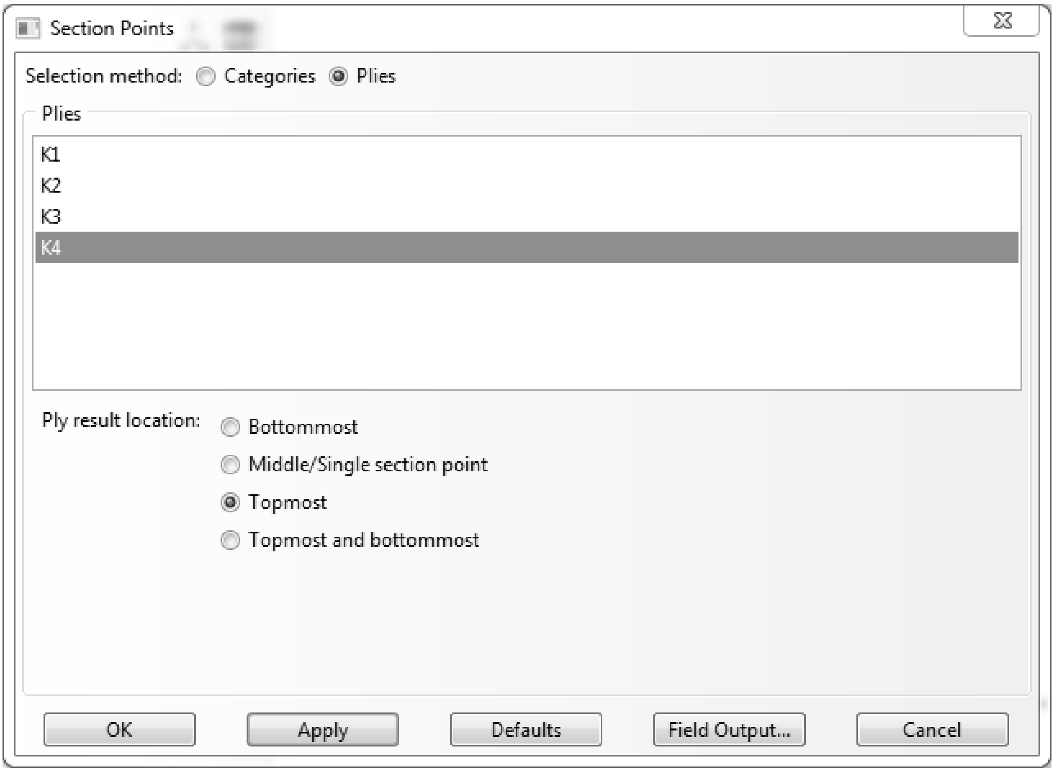
Thickness integration rule: ● Simpson ○ Gauss

□ Symmetric layers

Material	Thickness	Orientation Angle	Integration Points	Ply Name
Material-1	0.75	90	3	k1
Material-1	0.75	0	3	k2
Material-1	0.75	0	3	k3
Material-1	0.75	90	3	k4

Options:

Fig. 3.12: The Edit Section window is used to define the LSS.



Section Points

Selection method: ○ Categories ● Plies

Plies

K1

K2

K3

K4

Ply result location:
○ Bottommost
○ Middle/Single section point
● Topmost
○ Topmost and bottommost

Fig. 3.13: Section Points are used to select visualization at top and/or bottom of each lamina.

When the results are later visualized, one has to select where to visualize them. That is, one has to select the lamina and the position through the thickness of the lamina. This is done by invoking the `Section Points` *window, as follows:*

```
Module: Visualization
Menu:   Results, Section Points, Selection method: [Plies]
```

In the `Section Points` *window (Figure 3.13),* `K4, topmost` *selects results at the top surface of the laminate. Similarly,* `K1, bottommost` *selects the bottom surface of the laminate.*

The following pseudo code illustrates the whole procedure. If in doubt, open `Ex_3.5.cae` *available in [5]. For the sake of space, the materials and sections are created by running .py scripts, but they could be easily defined by using the methods shown in previous examples.*

```
# Units: N, mm, MPa
Menu:   File, Save as [c:\simulia\user\Ex_3.5\Ex_3.5.cae]
Menu:   File, Set Work Directory [c:\simulia\user\Ex_3.5]
```

i. Creating parts

```
Module: Part
Menu:   Part, Manager
    Create, [Part-A], 3D, Deformable, Shell, Planar,
    Approx size [500], Cont
Menu:   Add, Line, Rectangle, [0,0], [60,50], X, Done
    Create, [Part-AB], 3D, Deformable, Shell, Planar,
    Approx size [500], Cont
Menu:   Add, Line, Rectangle, [60,0], [90,50], X, Done
    Create, [Part-B], 3D, Deformable, Shell, Planar,
    Approx size [500], Cont
Menu:   Add, Line, Rectangle, [90,0], [120,50], X, Done
    # close the Part Manager
```

ii. *Creating assembly*

```
Module: Assembly
Menu:   Instance, Create, [Part-A],
    Independent,  # uncheck Auto-offset, OK
Menu:   Instance, Create, [Part-AB],
    Independent,  # uncheck Auto-offset, OK
Menu:   Instance, Create, [Part-B],
    Independent,  # uncheck Auto-offset, OK
Menu:   Instance, Merge/Cut, Intersecting Boundaries: Retain, Cont
    # select all, Done, # creates Part-1
    # on the .mdb tree, expand Assembly, expand Instances,
    right-click Part-1, Make Independent
```

iii. *Creating materials, section, and assigning sections to parts*

```
Module: Property
# to display the whole Part-1, set the tabbed menus as follows:
Module: [Property], Model: [Model-1], Part: [Part-1]
# create materials by running .py scripts
Menu:   File, Run Script, [AS4D--9310.py], OK
# create sections by running .py scripts
Menu:   File, Run Script, [laminate-A.py], OK
Menu:   File, Run Script, [laminate-AB.py], OK
Menu:   File, Run Script, [laminate-B.py], OK
# recall that from bottom to top, section A is [90/0/0/90],
# section B is [90/0], and section AB is [90/0/0]
Menu:   Section, Assignment Manager
    Create, # pick the region/part Section-A, Done
Section: [Section-A], Shell Offset Definition: [Bottom Surface], OK
    Create, # pick the region/part Section-AB, Done
Section: [Section-AB], Shell Offset Definition: [Bottom Surface], OK
    Create, # pick the region/part Section-B, Done
Section: [Section-B], Shell Offset Definition: [Bottom Surface],
    OK # close the Section Assignment Manager,
    Done # close current command
```

iv. *Defining analysis steps*

```
Module: Step
Menu:   Step, Create, Name [Step-1], Static/General, Cont, OK
Menu:   Output, Field Output Request, Edit, F-Output-1
    Output at shell, Specify [1,2,3,4,5,6,7,8,9,10,11,12], OK
```

v. Adding BC and loads

```
Module: Load
Menu:   BC, Manager
    Create, [BC-1], Step: Initial, Mechanical, Symm/Anti/Enca, Cont
    # pick edge @ x=0, Done, Encastre, OK
    Create, [BC-2], Step: Initial, Mechanical, Symm/Anti/Enca, Cont
    # pick the 3 segments @ y=0, Done, YSYMM, OK, # close BC Manager
    # since the part is a Shell, an edge load is: Shell edge load
    # load applies on the offset defined during Assign
Menu:   Load, Create
    [Load-1], Step: Step-1, Mechanical, Shell edge load, Cont
    # pick the edge at x=120, Done, Magnitude [-10], OK
```

vi. Creating mesh

```
Module: Mesh
Menu:   Seed, Instance, Approx global size: [5], Apply, OK
Menu:   Mesh, Controls, # select all the regions, Done
    Element Shape [Quad], Technique [Structured]
Menu:   Mesh, Element Type, # select all the regions, Done
    Standard, Quadratic, DOF [6], OK # element selected S8R
    Mesh, Instance, Yes
```

vii. Solving and visualizing the results

```
Module: Job
Menu:   Job, Manager
    Create, Cont, OK, Submit # when completed, Results

Module: Visualization
Menu:   Viewport, Viewport Annotation Options,
    Tab: State Block, Set Font, Size [10], Apply to [checkmark all],
    OK, OK
Menu:   File, Save Options, OK # to save options for future sessions
Menu:   Plot, Undeformed Shape
Menu:   Plot, Allow Multiple Plot States
Menu:   Plot, Deformed Shape # you will see both shapes
Menu:   Plot, Contours, On Deformed Shape
    Field Output Toolbar, [Primary], [U], [U3]
    # read the maximum value from the legend on top left, U3=2.578 mm
    Field Output Toolbar, [Primary], [S], [S11]
    # next to select the top or bottom surface of each lamina
Menu:   Result, Section Points
    Selection method: [Plies], K1, Bottommost, Apply
    # repeat for the other plies
    Selection method: [Plies], K1, topmost, Apply
    # repeat for the other plies
    # a gray region means the ply is not modeled in that region
```

The maximum deflection is $w = 2.578\,mm$. The maximum values of stress are reported in Table 3.3.

Table 3.3: Stresses calculated in Example 3.5.

Ply #	z [mm]	$\theta[°]$	$\sigma_{11}[MPa]$
1 (bot)	0.00	90	1.233
1 (top)	0.75	90	0.6206
2 (bot)	0.75	0	73.17
2 (top)	1.50	0	6.596
3 (bot)	1.50	0	6.596
3 (top)	2.25	0	-19.15
4 (bot)	2.25	90	0.1907
4 (top)	3.00	90	0.4094

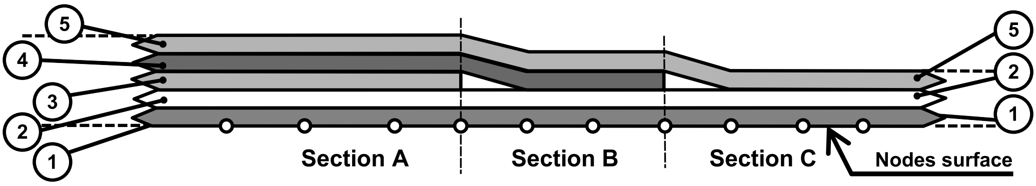


Fig. 3.14: Laminate with dropped laminas.

Example 3.6 *Model the laminate shown in Figure 3.14. The laminate in section A is a* $[+45/-45/0/90/0]$. *The thickness of each lamina is* 1.2 *mm. A ply drop-off ratio 1:10 is used. The strip is* 120 *mm long and* 100 *mm. It is loaded by tension* $N_x = 10$ *N/mm applied to the bottom edges on the strip. Use symmetry to model 1/2 of the tape. The material is AS4D/9310 (Table 3.1). Visualize and report the maximum value of transverse deflection* U_3. *Also, visualize and tabulate the maximum value of* σ_{11} *at the top surface of every lamina (k1, k2, k3, k4, k5). Note the absence of results for the laminas dropped in the region where those laminas have been dropped.*

Solution to Example 3.6 *In this example, we illustrate the use of partitions to separate the three regions. With a drop-off ratio 1:10, the transition length is* $2 \times 1.2 \times 10 = 24$ *mm, with drops at 12 mm on each side of the center line. With the center line at* 60 *mm from the left edge, the transitions are located at 48 mm and 72 mm. The bottom lamina is designated as lamina #1. Additional laminas are stacked from bottom to top in the positive normal direction of the element coordinate system. The dropped laminas are #3 and #4. The following pseudo code illustrates the whole procedure. If in doubt, open* `Ex_3.6.cae` *available in [5].*

```
# Units: N, mm, MPa
Menu:   File, Set Work Directory [c:\simulia\user\Ex_3.6]
Menu:   File, Save As [Ex_3.6.cae]
```

i. Creating parts

```
Module: Part
Menu:   Part, Create
    [Part-1], 3D, Deformable, Shell, Planar, Approx size [300], Cont
Menu:   Add, Line, Rectangle, [0,0], [120,50], X, Done
# creating 3 partitions for the different regions
```

```
Menu:   Tool, Datum, Plane, Offset from principal plane
    [YZ Plane], [48], Enter, [YZ Plane], [72], Enter, X
    # close the Create Datum pop-up window
Menu: Tools, Partition, Face, Use datum plane,
    # pick a datum plane, Create Partition
    # pick face to partition (area not partitioned yet), Done
    # pick datum plane, Create Partition, Done
    # close the Create Partition pop-up window
```

ii. *Creating materials, sections, and assigning sections to parts*

```
Module: Property
Menu:   Material, Create
    Name [AS4D--9310], Mechanical, Elasticity, Elastic, Type [Lamina]
    # you can cut and paste an entire row of values from Excel
    [1.3386E5, 7.706E3, 0.301, 4.306E3, 4.306E3, 2.76E3], OK
    # or use a .py file to create materials as in previous example
Menu:   Section, Manager
    Create, Name [Section-A], Shell, Composite, Cont
    # insert rows with right click until you have 5 rows
    Material [AS4D--9310] # for all, Thickness [1.2] # for all
    Orientation [45,-45,0,90,0], Integration points [3,3,3,3,3]
    Ply names [k1,k2,k3,k4,k5], OK
    Copy, [Section-AB], OK, Edit, # delete the center ply (k3), OK
    Copy, [Section-B],  OK, Edit, # delete the center ply (k4), OK
    # close Section Manager
Menu:   Section, Assignment Manager
    Create, # pick the region on the left, Done
    Section [Section-A], Shell Offset [Bottom surface], OK
    Create, # pick the region in the middle, Done
    Section [Section-AB], Shell Offset [Bottom surface], OK
    Create, # pick the region on the right, Done
    Section [Section-B], Shell Offset [Bottom surface], OK, Done
    # close Section Assignment Manager
```

iii. *Creating assembly*

```
Module: Assembly
Menu:   Instance, Create, Independent, OK
```

iv. *Defining analysis steps*

```
Module: Step
Menu:   Step, Create
    Name [Step-1], Procedure type: General, Static/General, Cont, OK
Menu:   Output, Field Output Request, Edit, F-Output-1
    Specify [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15], OK
```

v. *Adding BC and loads*

```
Module: Load
Menu:   BC, Create
    Name [BC-1], Step: Initial, Mechanical, Symm/Anti/Enca, Cont
```

```
        # pick edge @ x=0, Done, Encastre, OK
    Menu:   BC, Create
        Name [BC-2], Step: Initial, Mechanical, Symm/Anti/Enca, Cont
        # pick 3 segments @ y=0, Done, YSYMM, OK
        # since the part is a Shell, an edge load is: Shell edge load
        # load applied with an offset defined during Assign
    Menu:   Load, Create
        Name [Load-1], Step: Step-1, Mechanical, Shell edge load, Cont
        # pick edge @ x=120, Done, Magnitude [-10], OK
```

vi. Creating Mesh

```
    Module: Mesh
    Menu:   Seed, Instance, Approx global size: [5], Apply, OK
    Menu:   Mesh, Controls, # select all regions, Done,
        Element Shape [Quad], Technique [Structured], OK
    Menu:   Mesh, Element Type, # select all regions, Done
        [Standard], [Quadratic], DOF [6], OK, # element used S8R
    Menu:   Mesh, Instance, Yes
```

vii. Solving and visualizing the results

```
    Module: Job
    Menu:   Job, Manager
        Create, Cont, OK, Submit, # when completed, Results
    Module: Visualization
    Menu:   Viewport, Viewport Annotation Options
        Tab: State Block, Set Font, Size [10],
        Apply to [checkmark all], OK, OK
    Menu:   File, Save Options, OK # to save options for future sessions
    Menu:   Plot, Undeformed shape
    Menu:   Plot, Allow Multiple Plot States
    Menu:   Plot, Deformed Shape # you will see both shapes
    Menu:   Plot, Material Orient., On Def. Shape # identify the 1-axis
    Menu:   Result, Section Points, [Plies], K1, Bottommost
        # pick one ply at a time and comment on the results, OK
        Field Output Toolbar, [Primary], [S], [S11]
    Menu:   Result, Section Points, [Plies], K1, Topmost
        # pick one ply at a time and tabulate the results, OK
```

The maximum transverse deflection is 0.6075 *mm. The maximum values of stress are given in Table 3.4.*

Table 3.4: Maximum values of stress at the top surface of each ply for Example 3.6.

Ply #	z [mm]	$\theta[^\circ]$	$\sigma_{11}[MPa]$
1	1.2	45	10.1
2	2.4	-45	17.88
3	3.6	0	17.63
4	4.8	90	4.303
5	6.0	0	9.632

3.2.13 FEA of Sandwich Shells

Laminates should be modeled as *sandwich shells* when specifically designed for sandwich construction with thin faceplates and a thick, relatively weak, core (e.g., honeycomb, foam, balsa wood). The faceplates are intended to carry all, or almost all, of the bending and in-plane normal load. Conversely, the core is assumed to carry all of the transverse shear. Example 3.7 shows how to define and calculate a sandwich cantilever beam.

The following assumptions are customarily made for a sandwich shell:

- The terms H_{ij} in (3.9) depend only on the middle layer (core) and they can be calculated as
$$H_{ij} = \left(\overline{Q}^{*}_{ij}\right)_{core} t_{core} \quad ; \quad i,j = 4,5 \tag{3.17}$$
- The transverse shear moduli (G_{23} and G_{13}) are set to zero for the top and bottom layers (non-core layers) so they do not add artificial stiffness to the sandwich shell.
- The transverse shear strains and stresses in the face plate (non-core) layers are neglected or assumed to be zero.
- The transverse shear strains and shear stresses in the core are assumed constant through the thickness.

Example 3.7 *Consider a sandwich plate clamped around the boundary and loaded with uniform pressure on the top surface.*[2] *The sandwich is made of two face plates and an inner core with properties, thicknesses, and pressure given in Table 3.5. Use symmetry conditions to model one quarter of the plate. Use a surface mesh of 6 by 6 elements. Compute the center deflection using four different methods:*

a. Conventional shell elements S4R.
b. Continuum shell elements SC8R spanning the entire thickness, with the laminate defined inside the element.
c. Continuum shell elements SC8R stacked through the thickness, one for each face and one or more for the core.
d. Continuum shell elements SC8R for the core and conventional shell elements S4R for the faces.

Solution to Example 3.7 *Here we solve the problem in four ways. The results are tabulated in Table 3.6. The stacked continuum shell model (c) and stacked continuum shell plus skins (d) are more flexible than (a) and (b) models because in (c, d) the FSDT constraint (i.e., normals remain straight) is enforced separately in the core and two faces/skins. Thus, each lamina (core and each face) is able to rotate independently of the other laminas. This means that each lamina achieves its own value of transverse shear deformation. Since the core has lower transverse shear moduli (G_{13},G_{23}), it can deform more. This additional*

[2] This example is modified from [16, Test R0031/3] by making the edges clamped rather than simply supported. The change is made because it is difficult to impose a simple support with continuum shell elements. Doing so requires to use constraint equations, which complicates the example too much at this stage.

Table 3.5: Data for Example 3.7.

Quantity	Units	Face	Core
E_1	MPa	68947.57	0.068948
E_2	MPa	27579.03	0.068948
ν_{12}		0.3000	0.0000
G_{12}	MPa	12927.67	0.068948
G_{13}	MPa	12927.67	209.6006
G_{23}	MPa	12927.67	82.73708
thickness	mm	0.7112	19.0500
half plate width	mm	127.00	
pressure	MPa	0.689476	
thickness modulus	MPa	689475.7	689475.7

flexibility impacts the transverse deflection and more notably the transverse shear strains and stresses. Increasing the number of elements through the thickness in the core does not change the results because the core is homogeneous (not laminated). Therefore, its normal deforms as a straight line regardless of how many elements are used to discretize its thickness. The conventional and single element continuum models yield similar results because the single, laminated SC8R element has the same kinematics as the conventional S4R (provided the transverse compressibility of SC8R is constrained by selecting a high value for the thickness modulus).

Continuum shell elements in Abaqus *need a thickness modulus* E_3 *and they cannot read it from the given material properties, even if* E_3 *is given in* `Engineering Properties`*, or it is implied for* `Isotropic` *(*$E_3 = E$*) and* `Lamina` *(*$E_3 = E_2$*). So, the user must remember to assign a thickness modulus in tab* `Shell Parameters` *of the Edit Composite Layup window (Figure 3.15). Since a shell is incompressible in the thickness direction, we chose a thickness modulus ten times the highest modulus of the face material,*[3] *i.e.,* $E_3 = 10 \times E_1 = 689475.7\ MPa$*. No thickness modulus is needed for conventional shell elements because they are intrinsically incompressible through the thickness.*

(a) Solution with conventional shell elements

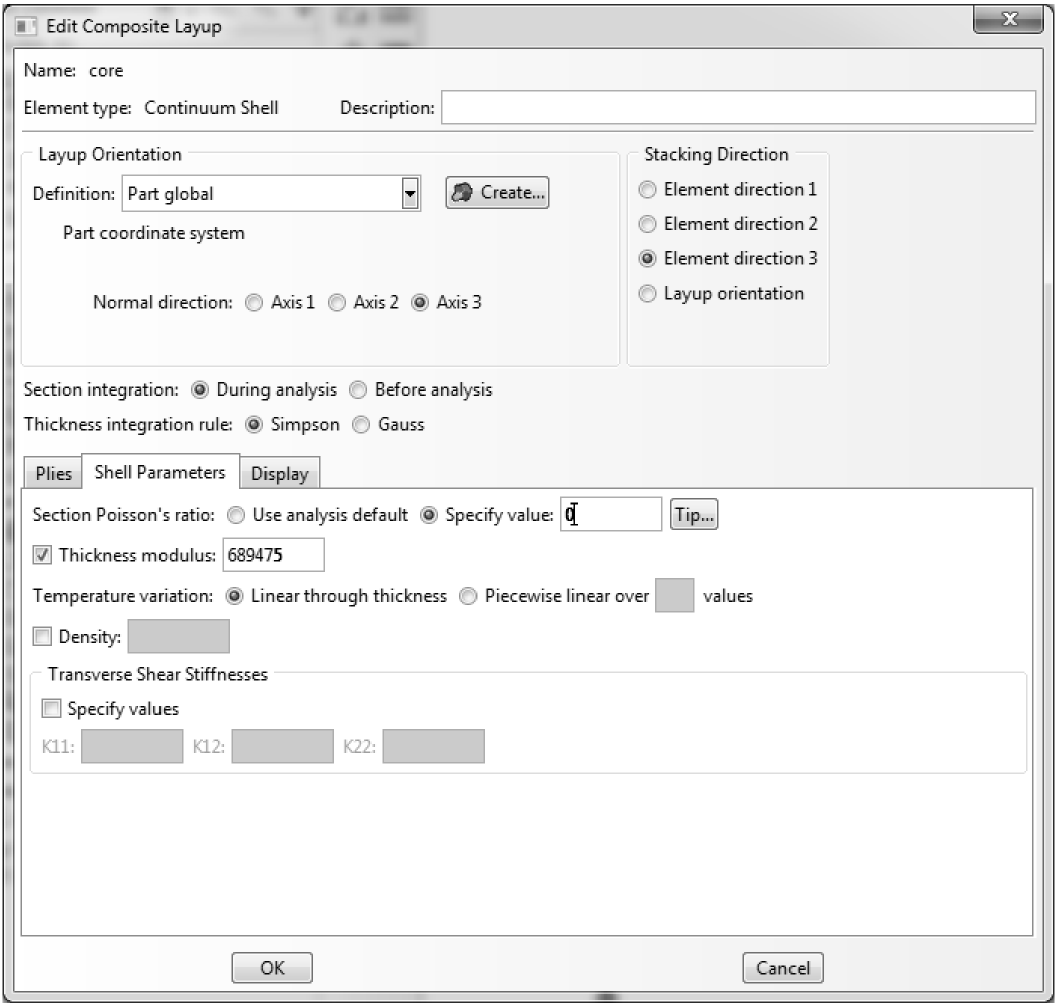
Here we show how to model the geometry of the sandwich plate as a 2D flat surface. Then, define the laminate inside the conventional shell element. The following pseudo code illustrates the whole procedure. If in doubt, open `Ex_3.7.a.cae` *available in [5].*

```
# NAFEMS sandwich clamped metric conventional shell elements
Menu:   File, Set Work Directory [c:\simulia\user\Ex_3.7]
Menu:   File, Save As [Ex_3.7.a.cae]
```

i. Creating parts

```
Module: Part
Menu:   Part, Create, 3D, Deformable, Shell, Planar, Cont
Menu:   Add, Line, Rectangle, [0,0], [127,127], X, Done
    # name the part to make it easier to use Composite Modeler later
Menu:   Tools, Set, Create, Name [plate], Cont, # select all, Done
```

[3] This is in agreement with r313_std_sc8r_composite.inp in [19, 4.9.3 R0031(3)].



Edit Composite Layup

Name: core

Element type: Continuum Shell Description:

Layup Orientation

Definition: Part global

Part coordinate system

Normal direction: ○ Axis 1 ○ Axis 2 ● Axis 3

Stacking Direction

○ Element direction 1

○ Element direction 2

● Element direction 3

○ Layup orientation

Section integration: ● During analysis ○ Before analysis

Thickness integration rule: ● Simpson ○ Gauss

Section Poisson's ratio: ○ Use analysis default ● Specify value: 0

☑ Thickness modulus: 689475

Temperature variation: ● Linear through thickness ○ Piecewise linear over values

□ Density:

Transverse Shear Stiffnesses

□ Specify values

K11: K12: K22:

Fig. 3.15: Specify a thickness modulus for the shell section.

ii. *Defining materials, sections, and assigning sections to parts*

```
Module: Property
Menu:   Material, Create
Name [face], Mechanical, Elasticity, Elastic, Type [Lamina]
    [68947.57, 27579.028, 0.3000, 12927.66938, 12927.66938,
     12927.66938], OK
Menu:   Material, Create
Name [core], Mechanical, Elasticity, Elastic, Type [Lamina]
    [0.06894757, 0.06894757, 0.0000, 0.06894757, 209.6006128,
    82.737084], OK
# use Composite Modeler instead of Create + Assign Sections
Menu:   Composite, Create, Name [NAFEMS-sandwich],
    Conventional Shell, Cont
    # Name, region, Material, Thickness, CSYS, Angle, Integ Points
    Ply-1, plate, face, 0.7112, <Layup>, 0, 3
    Ply-2, plate, core, 19.050, <Layup>, 0, 3
    Ply-3, plate, face, 0.7112, <Layup>, 0, 3
    OK
    # to assign Region double click the cell corresp. to desired ply
    Sets # to open Region Selection pop-up window, plate, Cont
    # to assign Material double click the cell corresp. to desired ply
    # in Select Material pop-up window, Material [face or core], OK
```

iii. *Creating assembly*

```
Module: Assembly
Menu:   Instance, Create, Independent, OK
```

iv. *Defining analysis steps*

```
Module: Step
Menu:   Step, Create
    Procedure type: Linear perturbation, Static/General, Cont, OK
```

v. *Adding BC and loads*

```
Module: Load
Menu:   Load, Create
    Name [Load-1], Step: Step-1, Mechanical, Pressure, Cont,
    # pick the part, Done, Choose a side for the shell [Brown]
    Magnitude [0.689476], OK
Menu:   BC, Manager
    Create, Step: Initial, Mechanical, Symm/Anti/Enca, Cont
    # pick edges @ x=0 and y=0, Done, Encastre, OK
    Create, Step: Initial, Mechanical, Symm/Anti/Enca, Cont
    # pick edge @ x=127, Done, XSYMM, OK
    Create, Step: Initial, Mechanical, Symm/Anti/Enca, Cont
    # pick edge @ y=127, Done, YSYMM, OK
    # close BC Manager
```

vi. *Meshing the model*

```
Module: Mesh
Menu:   Seed, Instance, Approx global size [21], OK
Menu:   Mesh, Controls, Element Shape [Quad], Tech [Structured], OK
Menu:   Mesh, Element Type, # select all surfaces, Done
    Element Library [Standard], Geometric Order [Linear], OK # S4R
Menu:   Mesh, Instance, Yes
```

vii. *Solving and visualizing the results*

```
Module: Job
Menu:   Job, Manager
    Create, Cont, OK, Submit # when completed, Results
Module: Visualization
    Field Output Toolbar, [Primary], [U], [U3]
```

The displacement at the upper-right corner is $-1.667\,mm$

(b) Solution with continuum shell elements

Here we show how to model the geometry of the sandwich plate as a 3D body. Then, define the laminate inside the continuum shell element. Note that in the composite modeler, the relative thicknesses do not need to add to 1.0. The actual ply thicknesses are calculated in terms of the mesh thickness. To emphasize this, we have chosen to enter the thicknesses of the plies in inches (0.028",075",0.028"), even though the model is set up in SI units. The following pseudo code illustrates the whole procedure. If in doubt, open `Ex_3.7.b.cae` *available in [5].*

```
# NAFEMS sandwich clamped metric laminated continuum shell elements
Menu:   File, Set Work Directory [c:\simulia\user\Ex_3.7]
Menu:   File, Save As [Ex_3.7.b.cae]
```

i. *Creating parts*

```
Module: Part
Menu:   Part, Create, 3D, Deformable, Solid, Extrusion, Cont
Menu:   Add, Line, Rectangle, [0,0], [127,127], X, Done
    Depth [20.4724], OK  # core + two faces
# name the part to make it easier to use Composite Modeler later
Menu:   Tools, Set, Create, Name [plate], Cont, # select all, Done
```

ii. *Defining materials, sections, and assigning sections to parts*

```
Module: Property
Menu:   Material, Create
    Name [face], Mechanical, Elasticity, Elastic, Type [Lamina]
    [68947.57, 27579.028, 0.3, 12927.66938, 12927.66938,
    12927.66938], OK
Menu:   Material, Create
    Name [core], Mechanical, Elasticity, Elastic, Type [Lamina]
    [0.06894757, 0.06894757, 0.0, 0.06894757, 209.6006128,
    82.737084], OK
# use Composite Modeler instead of Create + Assign Section
Menu: Composite, Create, Name [sandwich], Type: Continuum Shell, Cont
```

```
    # Name, region, Material, Thickness, CSYS, Angle, Integ Points
    Ply-1, plate, face, 0.028, <Layup>, 0, 3
    Ply-2, plate, core,  0.75, <Layup>, 0, 3
    Ply-3, plate, face, 0.028, <Layup>, 0, 3
    # to assign Region double click the cell corresp. to desired ply
    Sets # to open Region Selection pop-up window, plate, Cont
    # to assign Material double click the cell corresp. to desired ply
    # in Select Material pop-up window, Material [face or core], OK
    Tab: Shell Parameters
    Section Poisson's ratio: Specify value [0.0]
    # checkmark: Thickness modulus [689475.7], OK
    # the last two instructions are very important!
```

iii. *Creating assembly*

```
Module: Assembly
Menu:   Instance, Create, Independent, OK
```

iv. *Defining analysis steps*

```
Module: Step
Menu:   Step, Create
    Procedure type: Linear perturbation, Static/General, Cont, OK
Menu:   Output, Field Output Requests, Edit, F-Output-1
    Output variables: Edit variables [CF,CTSHR,LE,RF,S,SE,TSHR,U]
    Output at shell: Specify [1,2,3,4,5,6,7,8,9], OK
```

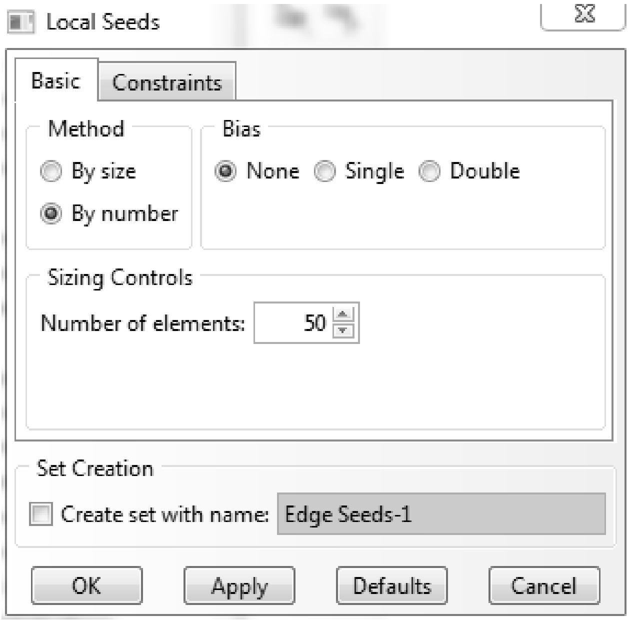
v. *Adding BC and loads*

```
Module: Load
Menu:   Load, Create
    Step: Step-1, Mechanical, Pressure, Cont
    # pick the top surface (z=20.4724), Done, Magnitude [0.689476], OK
Menu:   BC, Manager
    Create, Step: Initial, Mechanical, Symm/Anti/Enca, Cont
    # pick surfaces @ x=0 and y=0, Cont, Encastre, OK
    Create, Step: Initial, Mechanical, Symm/Anti/Enca, Cont
    # pick surface @ x=127, Cont, XSYMM, OK
    Create, Step: Initial, Mechanical, Symm/Anti/Enca, Cont
    # pick surface @ y=127, Cont, YSYMM, OK
    # close BC Manager
```

vi. *Meshing the model*

```
Module: Mesh
Menu:   Seed, Instance, Approx global size [21], OK
Menu:   Mesh, Controls, Element Shape [Hex], Technique [Sweep], OK
Menu:   Mesh, Element Type, # select all, Done
    Family [Continuum Shell], OK # notice the element chosen is SC8R
Menu:   Mesh, Instance, Yes
```

vii. *Solving and visualizing the results*



Local Seeds

Basic | Constraints

Method: ○ By size ● By number

Bias: ● None ○ Single ○ Double

Sizing Controls

Number of elements: 50

Set Creation

□ Create set with name: Edge Seeds-1

Fig. 3.16: Seeding an edge `By number`, using 50 elements through the length of the edge.

```
Module: Job
Menu:   Job, Manager
    Create, Cont, OK, Submit # when completed, Results
Module: Visualization
    Field Output Toolbar, [Primary], [U], [U3]
```

The displacement at the upper-right corner is -1.667 *mm.*

(c) Solution with stacked continuum shell elements

Here we show how to model the geometry of the sandwich plate as a 3D body, partitioned into a core and two faces. Each region is then discretized with continuum shell (CS) elements. Discretizing with 3D solid elements would require a very fine mesh of the plate dimensions (3000×600*) to avoid locking. In fact, the element dimensions could not be smaller than 10 times the thickness of the Aluminum face. Also in this example we show how to seed with a sweep and how to re-seed an edge, using the local seeds window (Figure 3.16). Remember to assign a thickness modulus in tab* `Shell Parameters` *of the Edit Composite Layup window (Figure 3.15). The following pseudo code illustrates the whole procedure. If in doubt, open* `Ex_3.7.c.cae` *available in [5]. Remeshing to accommodate 4 elements through the thickness of the core does not alter the results.*

```
# NAFEMS sandwich clamped metric stack 3 continuum shell elements
Menu:   File, Set Work Directory [c:\simulia\user\Ex_3.7]
Menu:   File, Save As [Ex_3.7.c.cae]
```

i. Creating parts

```
Module: Part
```

```
Menu:   Part, Create, 3D, Deformable, Solid, Extrusion, Cont
Menu:   Add, Line, Rectangle, [0,0], [127,127], X, Done
    Depth [20.4724], OK  # core + two faces
# partition the faces from the core
Menu:   Tool, Datum, Plane, Offset from principal plane
    [XY], [0.7112], Enter, [XY], [19.7612], Enter,
    X, # close Create Datum
Menu:   Tools, Partition, Cell, Use datum plane
    # the entire part is automatically selected
    Select datum plane # back datum plane, Create Partition
    # pick front cell to create the new partition, Done
    Select datum plane, Create Partition, Done
```

ii. Defining materials, sections, and assigning sections to parts

```
Module: Property
Menu:   Material, Manager
    Create, Name [face], Mechanical, Elasticity, Elastic,
    Type [Lamina]
    [68947.57, 27579.028, 0.3, 12927.66938, 12927.66938,
    12927.66938], OK
    Create, Name [core], Mechanical, Elasticity, Elastic,
    Type [Lamina]
    [0.06894757, 0.06894757, 0.0, 0.06894757, 209.6006128,
    82.737084], OK
    # close Material Manager, # zoom in to pick the regions easily
# use Composite Modeler instead of Create + Assign Section
Menu:   Composite, Create
    Name [front face], Initial ply count [1],
    Element Type [Continuum Shell], Cont
    # Name, region, Material, Rel. Thick., CSYS, Angle, Integ. Points
    Ply-1, <picked>, face, 1, <Layup>, 0, 3
    Tab: Shell Param., Section Poisson's ratio: Specify value [0.0]
    # checkmark: Thickness modulus: [689475.7], OK
    # the last two instructions are very important!
Menu:   Composite, Create
    Name [back face], Initial ply count [1],
    Element Type [Continuum Shell], Cont
    # Name, region, Material, Rel. Thick., CSYS, Angle, Integ. Points
    Ply-1, <picked>, face, 1, <Layup>, 0, 3
    Tab: Shell Param., Section Poisson's ratio: Specify value [0.0]
    # checkmark: Thickness modulus: [689475.7], OK
    # the last two instructions are very important!
Menu:   Composite, Create
    Name [core], Initial ply count [1], Type [Continuum Shell], Cont
    # Name, region, Material, Rel. Thick., CSYS, Angle, Integ. Points
    Ply-1, <picked>, core, 1, <Layup>, 0, 3
    Tab: Shell Param., Section Poisson's ratio: Specify value [0.0]
    # checkmark: Thickness modulus: [689475.7], OK
```

iii. Creating assembly

```
Module: Assembly
Menu:   Instance, Create, Independent, OK
```

iv. *Defining analysis steps*

```
Module: Step
Menu:   Step, Create
    Procedure type: Linear perturbation, Static/Linear pert., Cont, OK
Menu:   Output, Field Output Requests, Edit, F-Opuput-1
    Output variables: Edit variables [CF,CTSHR,LE,RF,S,SE,TSHR,U]
    Output at shell: Specify [1,2,3], OK
```

v. *Adding BC and loads*

```
Module: Load
Menu:   Load, Create
    Step: Step-1, Mechanical, Pressure, Cont
    # pick the top surface of the part (z=20.4724), Done
    Magnitude [0.689476], OK
Menu:   BC, Manager
    Create, Step: Initial, Mechanical, Symm/Anti/Enca, Cont
    # pick 3 surfaces @ x=0 and the 3 @ y=0, Done, Encastre, OK
    Create, Step: Initial, Mechanical, Symm/Anti/Enca, Cont
    # pick the three surfaces @ x=127, Done, XSYMM, OK
    Create, Step: Initial, Mechanical, Symm/Anti/Enca, Cont
    # pick the three surfaces @ y=127, Done, YSYMM, OK
    # close the BC Manager pop-up window
```

vi. *Meshing the model*

```
Module: Mesh
Menu:   Seed, Instance, Approx global size [21], OK
Menu:   Seed, Edges
    # select the four vertical edges of the core part, Done
    Method [By number], Number of elements [1], OK, X
Menu:   Mesh, Controls, # select all regions, Done, Sweep, OK
Menu:   Mesh, Element Type, # select all regions, Done
    Family [Continuum Shell], OK # SC8R
Menu:   Mesh, Instance, Yes
```

vii. *Solving and visualizing the results*

```
Module: Job
Menu:   Job, Manager
    Create, Cont, OK, Submit # when completed, Results
    Module: Visualization
    Field Output Toolbar, [Primary], [U], [U3]
```

The displacement at the upper-right corner is -1.892 *mm. For a more refined mesh in the core part of the sandwich, modify the procedure above as follows:*

```
Module: Mesh
Menu:   Seed, Edges
    # select the four vertical edges of the core part, Done
    Method [By number], Number of elements [4], OK, X
Menu:   Mesh, Instance, Yes
```

(d) Solution with conventional and continuum shell elements

Here we show how to model the geometry of the core as a 3D body, then apply two skins to represent the faces. The core is discretized with continuum shell elements and the faces with conventional shell elements. One must be careful to offset the location of the reference surface of the skins to coincide with the thickness of the core. Otherwise the results may be more or less wrong depending on how thick is the skin. To offset correctly, one has to place the reference at the top or bottom of the skin. This depends on the direction of the normal to the shell surface. Skins inherit the normal of the underlying shell. So, one has to visualize the normal to decide. If the normal is outward, then assigning the reference surface to the bottom of the skin will set the reference surface exactly where the surface nodes of the core are. The following pseudo code illustrates the whole procedure. If in doubt, open `Ex_3.7.d.cae` *available in [5].*

```
# NAFEMS sandwich clamped metric using skins
Menu:   File, Set Work Directory [c:\simulia\user\Ex_3.7]
Menu:   File, Save As [Ex_3.7.d.cae]
```

i. *Creating parts*

```
Module: Part
Menu:   Part, Create, 3D, Deformable, Solid, Extrusion, Cont
Menu:   Add, Line, Rectangle, [0,0], [127,127], X, Done
    Depth [19.05], OK  # notice you created only the core part
# name the part to make it easier to use Composite Modeler later
Menu:   Tools, Set, Create, Name [plate], Cont, # select all, Done
```

ii. *Defining materials, sections, and assigning sections to parts*

```
Module: Property
Menu:   Special, Skin, Create, # pick top face, Done
Menu:   Special, Skin, Create, # pick back face, Done, X
Menu:   Material, Create
    Name [face], Mechanical, Elasticity, Elastic, Type [Lamina]
    [68947.57, 27579.028, 0.3, 12927.66938, 12927.66938,
    12927.66938], OK
Menu:   Material, Create
    Name [core], Mechanical, Elasticity, Elastic, Type [Lamina]
    [0.06894757, 0.06894757, 0.0, 0.06894757, 209.6006128,
    82.737084], OK
# Notice the type of element used for the core (Continuum Shell)
Menu:   Composite, Create
    Name [core], Initial ply count [1], Type [Continuum Shell], Cont
    # Name, Region, Mat., Relat. Thick., CSYS, Angle, Integ. Points
    Ply-1, plate, core, 1, <Layup>, 0, 3
    Tab: Shell Parameters,
    Section Poisson's ratio: Specify value [0.0]
    # checkmark: Thickness modulus: [689475.7], OK
# Notice type of element used for skins/faces (Conventional Shell)
```

```
Menu:   Composite, Create
    Name [back face], Initial ply count [1], Type [Conv. Shell], Cont
    # Name, Region, Material, Thick., CSYS, Angle, Integ. Points
    Ply-1, <picked>, face, 0.7112, <Layup>, 0, 3
    # use actual thickness of the laminate
    Tab: Offset,
    Shell Reference Surface and Offsets [Bottom surface], OK
Menu:   Composite, Create, Name [front face],
    Initial ply count [1], Type [Conventional Shell], Cont
    # Name, Region, Material, Thick., CSYS, Angle, Integ. Points
    Ply-1, <picked>, face, 0.7112, <Layup>, 0, 3
    # use the actual thickness of the laminate
    Tab: Offset,
    Shell Reference Surface and Offsets [Bottom surface], OK
```

iii. *Creating assembly*

```
Module: Assembly
Menu:   Instance, Create, Independent, OK
```

iv. *Defining analysis steps*

```
Module: Step
Menu:   Step, Create
    Procedure type: Linear perturbation, Static/Linear pert., Cont, OK
Menu:   Output, Field Output Requests, Edit, F-Opuput-1
    Output variables: Edit variables [CF,CTSHR,LE,RF,S,SE,TSHR,U]
    Output at: Specify [1,2,3,4,5,6,7,8,9], OK  # important step!
```

v. *Adding BC and loads*

```
Module: Load
Menu:   Load, Create
    Step: Step-1, Mechanical, Pressure, Cont
    # pick the top surface of the part (z=19.05), Done
    Magnitude [0.689476], OK
Menu:   BC, Manager
    Create, Step: Initial, Mechanical, Symm/Anti/Enca, Cont
    # pick the surfaces @ x=0 and @ y=0, Done, Encastre, OK
    Create, Step: Initial, Mechanical, Symm/Anti/Enca, Cont
    # pick the surface @ x=127, Done, XSYMM, OK
    Create, Step: Initial, Mechanical, Symm/Anti/Enca, Cont
    # pick the surface @ y=127, Done, YSYMM, OK
    # close BC Manager pop-up window
```

vi. *Meshing the model*

```
Module: Mesh
Menu:   Seed, Instance, Approx global size [21], OK
Menu:   Mesh, Controls, Sweep, OK
Menu:   Mesh, Element Type, # pick the core part, Done
    Family [Continuum Shell], OK # element SC8R was applied
    # pick the front and back faces (the skins), Done
```

Table 3.6: Center Deflection for Example 3.7.

Method	Center deflection [mm]
Conventional	1.667
Single continuum	1.667
Stacked continuum	1.892
Continuum + skins	1.892

```
    Family [Shell], OK # element S4R was applied
Menu:   Mesh, Instance, Yes
```

vii. Solving and visualizing the results

```
Module: Job
Menu:   Job, Manager
    Create, Cont, OK, Submit # when completed, Results
Module: Visualization
    Field Output Toolbar, [Primary], [U], [U3]
```

The displacement at the upper-right corner is $-1.892\,mm$.

3.2.14 Element Coordinate System

In the pre-processor, during the definition of the laminate, it is very important to know the orientation of the laminate coordinate system. Material properties, the relative lamina orientation with respect to the laminate axis, and other parameters and properties are defined in the laminate coordinate system, unless specified otherwise. Also, it can be used to obtain the derived results (strains and stress) in these directions. In FEA, the laminate coordinate system is associated to the element coordinate system, with a unique right-handed orthogonal system associated to each element.

The element coordinate system orientation is associated with the element type. For bar or beam elements the orientation of the x-axis is generally along the line defined by the end nodes of the element. For solid elements in two and three dimensions, the orientation is typically defined parallel to the global coordinate system. For shell elements, axes x and y are defined on the element surface, with the z-axis always normal to the surface. The default orientation of x and y axes depends on the commercial code and the element type.

There are various ways to define the default orientation of the *element coordinate system* in shell elements. In *ANSYS* (Figure 3.17.(a)), the x-axis is aligned with the edge defined by the first and second nodes of each element, the z-axis normal to the shell surface (with the outward direction determined by the right-hand rule), and the y-axis perpendicular to the x- and z-axis. In *MSC-MARC*, the x-axis is aligned with a line joining the middle points of two opposite edges (Figure 3.17.(b)). In *Abaqus* (Figure 3.17.(c)), the local x-direction is calculated projecting the global X-direction onto the surface of the element, i.e.,

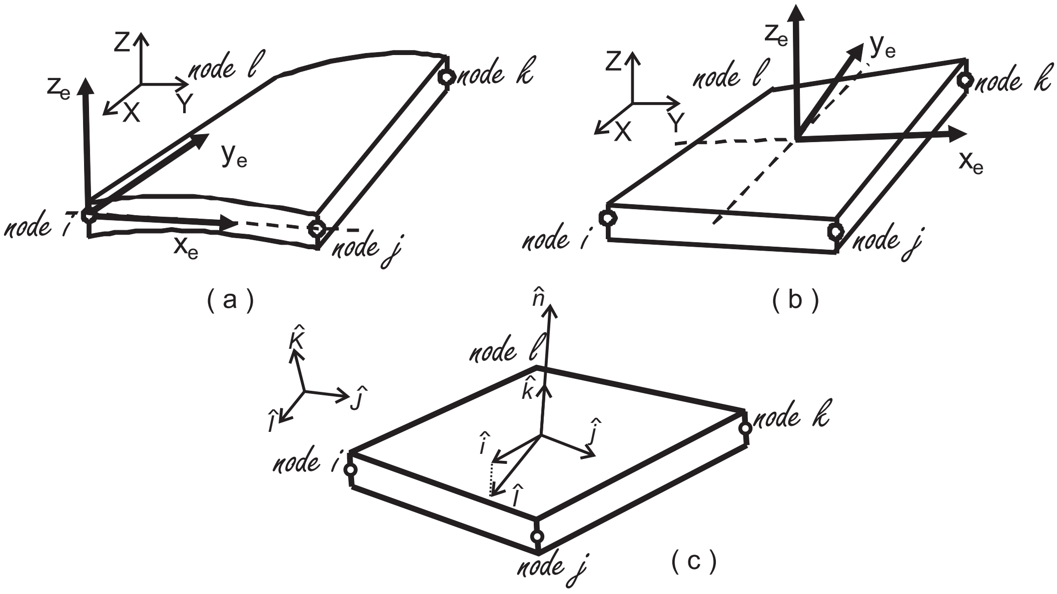


Fig. 3.17: Default orientations of element coordinate systems in shell elements: (a) *ANSYS*, (b) *MSC-MARC*, (c) *Abaqus*.

$$
\begin{aligned}
\hat{\imath} &= (\hat{I} - \hat{\imath}^*)/\|\hat{I} - \hat{\imath}^*\| \\
\hat{k} &= \hat{n} \\
\hat{\jmath} &= \hat{k} \times \hat{\imath}
\end{aligned}
\tag{3.18}
$$

with $\hat{\imath}^* = (\hat{I} \cdot \hat{n})\hat{n}$ being the projection of the global x-direction, $\hat{I}$, along the normal of the element, $\hat{n}$. If the global X-direction, $\hat{I}$, is within 0.1° of the normal to the element, $\hat{n}$, i.e., they are almost parallel, the local x-direction is calculated using the global z-direction, $\hat{K}$, instead of $\hat{I}$. The local coordinate system can be redefined using the procedures described in the *Abaqus* documentation [7, Section 2.2.5].

If no additional rotation is specified, the *laminate coordinate system* coincides with the element coordinate system. Additional rotations are specified by `ESYS` in *ANSYS* and `CSYS` in *Abaqus*, as shown in Examples 3.8 and 3.9 (pp. 151, 153).

In Examples 3.2-3.7, only rectangular plates with rectangular elements are analyzed. All of them have the first and the second node aligned with the global X-axis. Therefore, the material axes have been chosen parallel to the global axis. But this doesn't need to be the case. Most commercial codes have utilities to change the element coordinate system. Example 3.8 illustrates how to define a local coordinate system. Example 3.9 illustrates how it can be done in a shell with curvature. Example 3.10 illustrates how different orientations can be used in different locations of the structure.

Example 3.8 *Use a local coordinate system to model the plate of Example 3.2 (p. 113) if* *the orthotropic material is rotated +30 degrees with respect to the x-direction. This example continues with Example 3.13, p. 168.*

Solution to Example 3.8 *This example is a continuation of Example 3.2 (p. 113). However, the material is no longer symmetric so we cannot reuse* `Ex._3.2.cae` *and we must construct a new full model. In the procedure presented below the mechanical properties of the material are entered using the definition of a* `Lamina` *instead of using the* `[ABDH]` *matrix.*

```
Menu: File, Set Work Directory, [C:\SIMULIA\User\Ex_3.8], OK
Menu: File, Save as, [Ex_3.8.cae], OK
```

i. *Creating parts*

```
Module: Part
Menu:   Part, Create
    3D, Deformable, Shell, Planar, Approximate size [4000], Cont
Menu:   Add, Line, Rectangle
    # Enter the points [-2000,-1000], [2000,1000], X, Done
```

ii. *Defining materials*

```
Module: Property
Menu:   Material, Create, Mechanical, Elast., Elastic, Type: Lamina
    [133860, 7706, 0.301, 4306, 4306, 2760], OK
```

iii. *Creating a local axis oriented at 30 deg and assigning material orientation to regions*

```
Menu:   Tools, Datum
    # enter: origin, point along new x-axis, point on xy-plane
    # cos(30)=0.866, sin(30)=0.5
    CSYS, 3 Points, Name [csys-1], Rectangular, Cont
    [0,0,0], [0.866,0.5,0], [1,1,0], # close pop-up windows
Menu:   Assign, Material Orientation, # pick the region, Done
    Datum CSYS List, Names: csys-1, OK, # look at the figure, OK
```

iv. *Creating sections and assigning sections to parts*

```
Menu:   Section, Create
    Shell, Homogeneous, Cont
    Section integration: Before analysis, Thickness: Value [10], OK
Menu:   Assign, Section, # pick part, Done, OK
```

v. *Creating assembly*

```
Module: Assembly
Menu:   Instance, Create, Independent, OK
    # the global and user-defined coordinate systems are displayed
```

vi. *Defining analysis steps*

```
Module: Step
Menu:   Step, Create, Name [Step-1], Static/General, Cont, OK
```

vii. *Adding BC and loads*

```
Module: Load
Menu:   BC, Manager
    Create, Name [SS1-xx], Step: Initial, Mech., Disp/Rota, Cont
    # pick the lines @ x=-2000 and x=2000, Done, U2, U3, RU1, OK
    Create, Name [SS1-yy], Step: Initial, Mech., Disp/Rota, Cont
    # pick the lines @ y=-1000 and y=1000, Done, U1, U3, RU2, OK
    # close BC Manager
Menu:   Load, Create
    Step [Step-1], Mech., Pressure, Cont, # pick the part, Done
    # Choose a side for the shell: Brown, Magnitude [0.12e-3], OK
```

viii. *Meshing the model*

```
Module: Mesh
Menu:   Seed, Instance, Approximate global size [500], Apply, OK
Menu:   Mesh, Controls, Elem. Shape: Quad, Technique: Structured, OK
Menu:   Mesh, Element Type, Geometric Order: Quadratic, DOF: 5, OK
Menu:   Mesh, Instance, Yes
```

ix. *Solving and visualizing the solution*

```
Module: Job
Menu:   Job, Manager
    Create, Cont, OK, Submit # when completed, Results
Module: Visualization
Menu:   Plot, Contours, On Deformed Shape
    Field Output Toolbox, Primary, U, U3
```

The maximum deflection of the model is 8.704 *mm.*

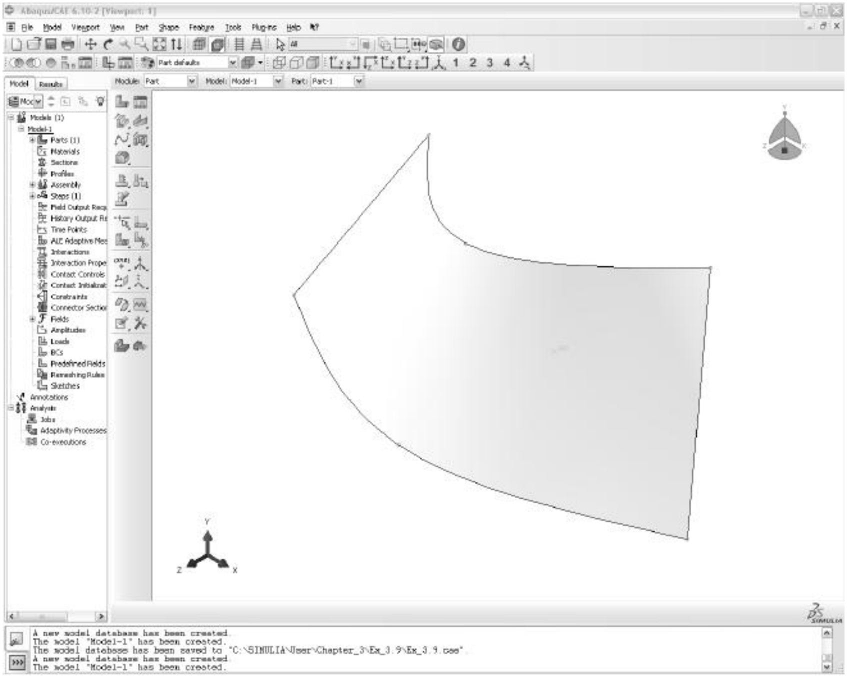
Example 3.9 *Align the element coordinate system (laminate coordinate system) of a 3D curved shell with the projection of the global Y-axis onto the surface of the shell.*

Solution to Example 3.9 *The material used is AS4D/9310 in a [90/45/-45] laminate configuration. This example is solved in two steps. First, the laminate coordinate system will be aligned with the element coordinate system by selecting* `CSYS: <Layup>` *in* `Composite Layup`*, The orientation of the element coordinate system is explained in Section 3.2.14. Second, an user-defined orientation will be created by a rotation of* 90° *around the z-axis of the element coordinate system, thus aligning the laminate coordinate system with the projection of the global Y-axis on the surface of the shell.*

```
Menu: File, Set Work Directory, [C:\SIMULIA\User\Ex_3.9], OK
Menu: File, Save as, [Ex_3.9.cae], OK
```

i. *Creating parts*

```
Module: Part
Menu:   Part, Create, 3D, Deformable, Point, Approx. size [500], Cont
    [0,0,0] # creates a reference point denoted by RF
Menu:   Tools, Datum, Point, Enter coordinates
    [300,0,135], [0,0,235], [-300,0,135], [200,200,0],
    [0,200,135], [-200,200,0], X, # close pop-up, Views: Iso View
Menu:   Shape, Wire, Spline, # pick the two rightmost points, Done
```



```
A new model database has been created.
The model "Model-1" has been created.
The model database has been saved to "C:\SIMULIA\User\Chapter_3\Ex_3.9\Ex_3.9.cae".
A new model database has been created.
The model "Model-1" has been created.
```

Fig. 3.18: Part geometry in Example 3.9.

```
Menu:   Shape, Wire, Spline, # leftmost and topmost points, Done
Menu:   Shape, Wire, Spline, # pick the lower three points, Done
Menu:   Shape, Wire, Spline, # pick the upper three points, Done
Menu:   Shape, Shell, Loft
    Insert Before, # pick the leftmost straight line, Done
    Insert After, # pick the rightmost straight line, Done
    Tab: Transition, Method: Select path
    Add, # pick lower curve, Done, Add, # pick upper curve, Done, OK
```

The part should look like the one displayed in Fig. 3.18.

ii. *Defining materials, section, and assigning sections to parts*

```
Module: Property
Menu:   Material, Create, Mechanical, Elast., Elastic, Type: Lamina
    [133860, 7706, 0.301, 4306, 4306, 2760], OK
Menu:   Tools, Datum, # create a CSYS to be used later
    CSYS, 3 Points, Name [user-csys], Rectangular, Cont
    [0,0,0], [0,1,0], [-1,1,0], # close pop-up windows
Menu:   Composite, Create, Element Type: Conventional Shell, Cont
    Region # double click the cell corresponding to each ply
    # pick the part, Done # the cell must read (Picked)
    Material # double click the cell corresponding to each ply, OK
    Thickness [1.05] # all plies, CSYS # <Layup> all plies
    Rotation Angle [90,45,-45], Integ. Points [3] # all plies, OK
```

iii. *Creating assembly*

```
Module: Assembly
Menu:   Instance, Create, Independent, OK
    # the global and user-defined coordinate systems are displayed
```

iv. *Defining analysis steps*

```
Module: Step
Menu:   Step, Create, Name [Step-1], Static/General, Cont, OK
Menu:   Output, Field Output Requests, Edit, F-Output-1
    Output at shell: Specify [1,2,3,4,5,6,7,8,9], OK
```

v. *Meshing the model*

```
Module: Mesh
Menu:   Seed, Instance, Approximate global size [100], Apply, OK
Menu:   Mesh, Controls, Element Shape: Quad, Technique: Structure, OK
Menu:   Mesh, Instance, Yes
```

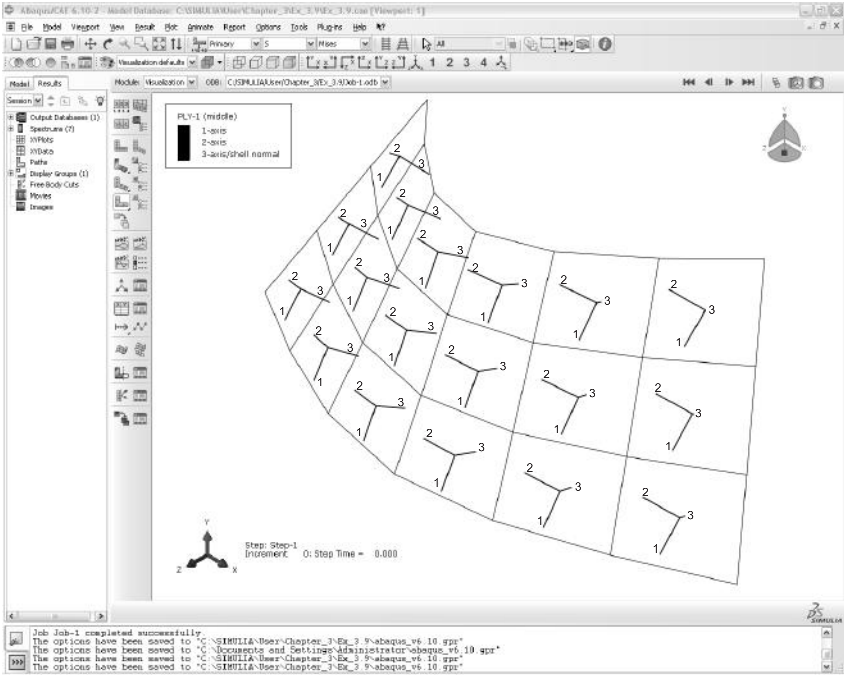
Since the goal of this example is to show the user how the element and laminate coordinate system are oriented, it is not necessary to apply loads, BC, or to solve the problem. However, doing a `Data Check` *of the model allows the user to visualize the orientation of the element coordinate systems.*

vi. *Visualizing laminate coordinate systems*

```
Module: Job
Menu:   Job, Manager
    Create, Cont, OK, Data Check # when completed, Results
# to visualize the coordinate system of the elements
Module: Visualization
Toolbar: Render Style, Render Model: Wireframe
Menu:   Plot, Material Orientations, On Undeformed Shape
# select different plies and visualize their orientations
Menu:   Results, Section Points, Selection method: Plies
```

By selecting `<Layup>` as the option for the `CSYS` of the `Composite Layup`, the user is choosing to define the laminate coordinate system aligned with the element coordinate system. To use an user-defined orientation, the `CSYS` in the `Composite Layup` must be changed. In the pseudocode for Example 3.9, an user-defined orientation was defined and named `user-csys`. The pseudocode below shows how to modify the problem to use an user-defined orientation.

```
# Defining materials, sections, and assigning sections to parts
Module: Property
Menu:   Composite, Edit, CompositeLayup-1
    Region # must read (Picked) for all the plies
    Material # must read Material-1 for all the plies
    Thickness [1.05] # for all plies
    CSYS # double the cell corresponding to each ply
    Base Orientation: CSYS, Select, Datum CSYS List, user-csys, OK, OK
    # must read user-csys.3 for all the plies
    Rotation Angle [90,45,-45], Integ.Points [3] # all plies, OK
```



Abaqus/CAE 6.10-2 - Model Database: C:\SIMULIA\User\Chapter_3\Ex_3.9\Ex_3.9.cae [Viewport: 1]

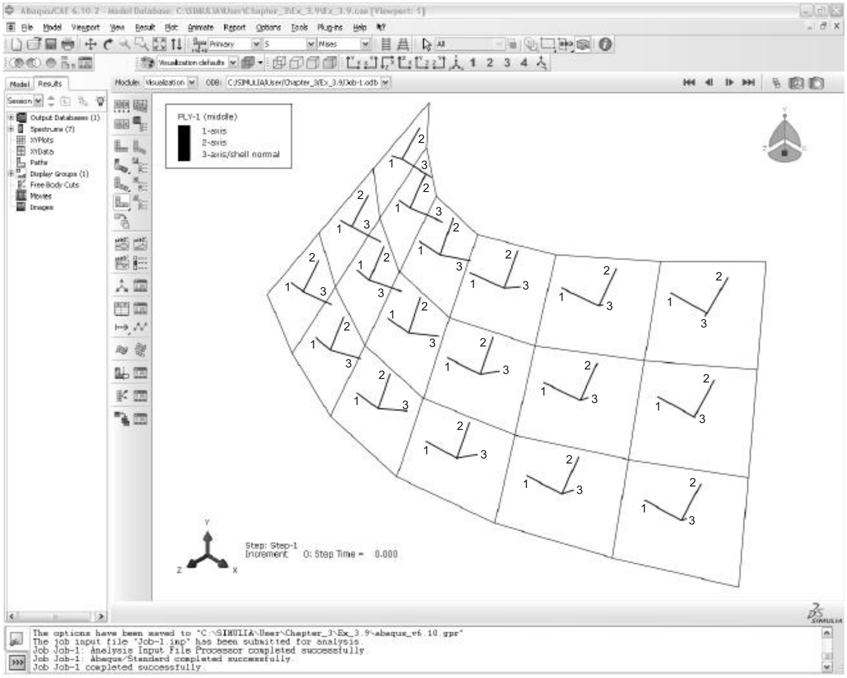
Module: Visualization ODB: C:/SIMULIA/User/Chapter_3/Ex_3.9/Job-1.odb

PLY-1 (middle)
1-axis
2-axis
3-axis/shell normal

Step: Step-1
Increment 0: Step Time = 0.000

```
Job Job-1 completed successfully.
The options have been saved to "C:\SIMULIA\User\Chapter_3\Ex_3.9\abaqus_v6.10.gpr"
The options have been saved to "C:\Documents and Settings\Administrator\abaqus_v6.10.gpr"
The options have been saved to "C:\SIMULIA\User\Chapter_3\Ex_3.9\abaqus_v6.10.gpr"
The options have been saved to "C:\SIMULIA\User\Chapter_3\Ex_3.9\abaqus_v6.10.gpr"
```

Fig. 3.19: Element coordinate system defined using Material Layup for the 90° layer of a [90/45/-45] laminate.



Abaqus/CAE 6.10-2 - Model Database: C:\SIMULIA\User\Chapter_3\Ex_3.9\Ex_3.9.cae [Viewport: 1]

Module: Visualization ODB: C:/SIMULIA/User/Chapter_3/Ex_3.9/Job-1.odb

PLY-1 (middle)
1-axis
2-axis
3-axis/shell normal

Step: Step-1
Increment 0: Step Time = 0.000

```
The options have been saved to "C:\SIMULIA\User\Chapter_3\Ex_3.9\abaqus_v6.10.gpr"
The job input file "Job-1.inp" has been submitted for analysis.
Job Job-1: Analysis Input File Processor completed successfully.
Job Job-1: Abaqus/Standard completed successfully.
Job Job-1 completed successfully.
```

Fig. 3.20: Element coordinate system defined using user-defined CSYS for the 90° layer of a [90/45/-45] laminate.

The lamina (ply) coordinate system for the 90° layer, before and after rotation, is shown in Figures 3.19 and 3.20. By visualizing the lamina coordinate system for various laminas, one can see how the coordinate systems are affected by the user-defined orientation.

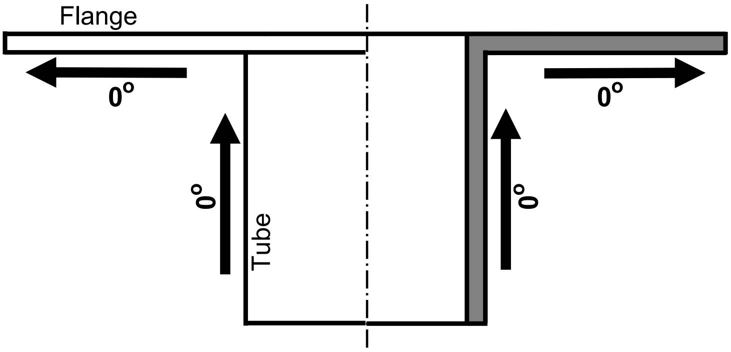


Fig. 3.21: Reference axis in a flange tube.

Example 3.10 *Model in Abaqus a flanged tube with axial and radial laminate orientation. In the cylindrical part, the reference axis will be in the longitudinal direction. In the flange, the reference axis will be radial (see Figure 3.21).*

Solution to Example 3.10 *The material used is AS4D/9310 in a [0/45/-45] laminate configuration. To satisfy the required orientations a cylindrical coordinate system will be used. The pseudocode below shows how to align the coordinate system of the elements of the cylinder in the axial direction (z-axis of the cylindrical coordinate system), and the coordinate system of the elements of the flange in the radial direction of the cylindrical coordinate system.*

```
Menu: File, Set Work Directory, [C:\SIMULIA\User\Ex_3.10], OK
Menu: File, Save as, [Ex_3.10.cae], OK
```

i. Creating parts

```
Module: Part
Menu:   Part, Create, 3D, Solid, Extrusion, Approx size [600], Cont
Menu:   Add, Circle, [0,0], [0,175], [0,0], [0,275], X, Done
    Depth [350], OK
Menu:   Shape, Shell, From Solid, # pick the part, Done, X
Menu:   Tools, Geometry Edit, Category: Face, Method: Remove
    # pick the external cylindrical surface, Done
    # pick the flat surface at the back of the model, Done
    X, # close the Geometry Edit pop-up window
Menu:   Tools, Set, Manager
    Create, Name [cylinder], Cont, # pick cylindrical surface, Done
    Create, Name [flange], Cont, # pick the flat surface, Done
    # close the Set Manager pop-up window, # sets will be used later
```

ii. Defining materials, section, and assigning sections to parts

```
Module: Property
Menu: Material, Create, Mechanical, Elasticity, Elastic, Type: Lamina
    [133860, 7706, 0.301, 4306, 4306, 2760], OK
Menu: Tools, Datum, CSYS, 3 Points, Name [cyl-csys], Cylindrical, Cont
    [0,0,0], [1,0,0], [1,1,0], # close pop-up windows
Menu:   Composite, Manager
    Create, Name [cyl-zone], Element Type: Conventional Shell, Cont
    Region # double click the cell corresponding to each ply
    Sets, cylinder, Cont, # the cell must read cylinder for all plies
    Material # double click the cell corresponding to each ply, OK
    # must read Material-1 for all plies, Thickness [1.05] # all plies
    CSYS # double click the cell corresponding to each ply
    Base Orientation: CSYS, Select, Datum CSYS List, cyl-csys, OK
    Normal Direction: Axis 2, OK, # must read cyl-csys.2 for all plies
    Rotation Angle [0,45,-45], Integration Points [3] # all plies, OK
    Create, Name [flange-zone], Element Type: Conventional Shell, Cont
    Region # double click the cell corresponding to each ply
    Sets, flange, Cont, # the cell must read flange for all plies
    Material # double click the cell corresponding to each ply, OK
    # must read Material-1 for all cells, Thickness [1.05] # all plies
    CSYS # double click the cell corresponding to each ply
    Base Orientation: CSYS, Select, Datum CSYS List, cyl-csys, OK
    Normal Direction: Axis 3, OK, # must read cyl-csys.3 for all plies
    Rotation Angle [0,45,-45], Integration Points [3] # all plies, OK
    # close Composite Layup Manager
```

iii. *Creating assembly*

```
Module: Assembly
Menu:   Instance, Create, Independent, OK
    # the global and user-defined coordinate systems are displayed
```

iv. *Defining analysis steps*

```
Module: Step
Menu:   Step, Create, Name [Step-1], Static/General, Cont, OK
Menu:   Output, Field Output Requests, Edit, F-Output-1
    Output at shell: Specify [1,2,3,4,5,6,7,8,9], OK
```

v. *Meshing the model*

```
Module: Mesh
Menu:   Seed, Instance, Approximate global size [75], Apply, OK
Menu:   Mesh, Controls, # pick all the surfaces, Done
    Element Shape: Quad, Technique: Free, OK
Menu:   Mesh, Instance, Yes
```

The goal of this example is to show the user how to align the element's orientation using a cylindrical coordinate system. Therefore, it is not necessary to apply loads, BC, or to solve the problem. However, doing a `Data Check` *of the model allows the user to visualize the orientation of the element coordinate systems.*

vi. *Visualizing coordinate element systems*

```
Module: Job
Menu:   Job, Manager
    Create, Cont, OK, Data Check # when completed, Results
Module: Visualization
    Toolbar: Render Style, Render Model: Shaded
    # to easily visualize the coordinate system of all the elements
Menu:   Plot, Material Orientations, On Undeformed Shape
Menu:   Results, Section Points, Selection method: Plies
    # select the different plies and visualize their orientations
```

3.2.15 Constraints

Constraints are used to restrict the motion of portions of a model in a variety of ways, as summarized below:

Tie Constraint tie two regions regardless of the meshes on the two regions being identical or not. See Example 6.3, p. 236.

Rigid Body Constraint couples the motion of regions in an assembly to the motion of a reference point. *Relative positions* of regions that make up the rigid body will remain unchanged during the analysis. Within rigid body constraint, `pin nodes` makes all displacements equal to those of the reference point, and `tie nodes` makes all displacements *and rotations* equal to those of the reference point. Rigid body constraints are very useful to force pinned boundary conditions as in Example 3.11.

Coupling Constraint couples the motion of a surface to the motion of a single point. Although a mesh of 3D solids has only 3 DOF per node, the reference point will have 6 DOF. Therefore, it will be necessary to constrain the rotations of the RP if the intent is to have the coupled surface translate but not rotate as a rigid surface. See Example 7.5, p. 269.

Multi-point Constraint (MPC) couples the motion of selected slave nodes to the motion of a single point.

Shell-to-solid Coupling Constraint couples the motion of a shell edge to the motion of an adjacent solid face. This is useful for detailed modeling of small regions using solid elements while large portions of the model are idealized with shell elements.

Embedded Region Constraint embeds a (small) region of a model within a host region of a (larger) model. This is useful for local-global analysis, multi-scale analysis, and multi-physics analysis.

Example 3.11 *Create a FE model for a pultruded composite column under axial compression load $P = 11,452$ N [20] and calculate the end axial displacement $u(L/2)$, where $x = 0$ is located at the half-length of the column. The column is simply supported (pinned) at both ends $x = (-L/2, L2)$ but only one half needs to be discretized because the model is symmetric at half the length of the column. Its length is $L = 1.816$ m. The cross section of the column* *is that of a wide-flange I-beam (also called H-beam) with equal outside height and width, $H = W = 304.8$ mm. The thickness of both the flange and the web is $t_f = t_w = 12.7$ mm. The material properties are given by the A, B, D, and H matrices, with units [mm MPa], [mm^2 MPa], [mm^3 MPa], and [mm MPa], respectively.*

For the flange:

$$[A] = \begin{bmatrix} 335{,}053 & 47{,}658 & 0 \\ 47{,}658 & 146{,}155 & 0 \\ 0 & 0 & 49{,}984 \end{bmatrix} \quad ; \quad [B] = \begin{bmatrix} -29{,}251 & -1{,}154 & 0 \\ -1{,}154 & -5{,}262 & 0 \\ 0 & 0 & -2{,}274 \end{bmatrix}$$

$$[D] = \begin{bmatrix} 4{,}261{,}183 & 686{,}071 & 0 \\ 686{,}071 & 2{,}023{,}742 & 0 \\ 0 & 0 & 677{,}544 \end{bmatrix} \quad ; \quad [H] = \begin{bmatrix} 34{,}216 & 0 \\ 0 & 31{,}190 \end{bmatrix}$$

For the web:

$$[A] = \begin{bmatrix} 338{,}016 & 44{,}127 & 0 \\ 44{,}127 & 143{,}646 & 0 \\ 0 & 0 & 49{,}997 \end{bmatrix} \quad ; \quad [B] = \begin{bmatrix} -6{,}088 & -14{,}698 & 0 \\ -14{,}698 & -6{,}088 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$[D] = \begin{bmatrix} 4{,}769{,}538 & 650{,}127 & 0 \\ 650{,}127 & 2{,}155{,}470 & 0 \\ 0 & 0 & 739{,}467 \end{bmatrix} \quad ; \quad [H] = \begin{bmatrix} 34{,}654 & 0 \\ 0 & 31{,}623 \end{bmatrix}$$

Solution to Example 3.11 *The geometry of the I-beam is defined by the midsurfaces of the web and the flanges. Notice that the geometry defined in the procedure below corresponds to an I section 292.1 mm in height and 304.8 in width. The column is symmetric about the longitudinal direction (Z-axis). Therefore, only one half of the length of the column is modeled. The load is applied as a concentrated load. This requires the use of a rigid body constraint of the type* `tie nodes` *on the loaded end to guarantee that all the points at the loading edge follow the displacements and rotations of the point where the load is applied.*

```
Menu: File, Set Work Directory, [C:\SIMULIA\User\Ex_3.11], OK
Menu: File, Save as, [Ex_3.11.cae], OK
```

i. Creating parts

```
Module: Part
Menu:   Part, Create
    3D, Deformable, Shell, Extrusion, Approx size [500], Cont
Menu:   Add, Line, Connected Lines, [-152.4,-146.05],
            [152.4,-146.05], X
Menu:   Add, Line, Connected Lines, [-152.4,146.05],
            [152.4,146.05], X
Menu:   Add, Line, Connected Lines, [0,-146.05],
            [0,146.05], X, Done
    Depth [908], OK
```

ii. Defining materials, section, and assigning sections to parts

```
Module: Property
Menu:   Section, Create, Name [flange],
            Shell, General Shell Stiffness, Cont
    # In the pop-up window fill the stiffness matrix as follows
    Tab: Stiffness
    # Upper part of matrix A in rows 1 to 3, columns 1 to 3
    # matrix B in rows 1 to 3, columns 4 to 6
    # Upper part of matrix D in rows 4 to 6, columns 4 to 6
    # The Stiffness matrix should be
    # 335053    47658       0     -29251     -1154         0
    #          146155       0      -1154     -5262         0
    #                   49984          0         0     -2274
    #                            4261183    686071         0
    #                                      2023742         0
    #                                                 677544
    Tab: Advanced
    # The values corresponding to the H matrix are entered here
    Transverse Shear Stiffness: # checkmark: Specify values
    K11 [34216], K12 [0], K22 [31190], OK
Menu:   Section, Create, Name [web],
            Shell, General Shell Stiffness, Cont
    Tab: Stiffness
    # The Stiffness matrix should be
    # 338016    44127       0      -6088    -14698         0
    #          143646       0     -14698     -6088         0
    #                   49997          0         0         0
    #                            4769538    650127         0
    #                                      2155470         0
    #                                                 739467
    Tab: Advanced
    Transverse Shear Stiffness: # checkmark: Specify values
    K11 [34654], K12 [0], K22 [31623], OK
Menu:   Assign, Section
    # pick the four regions forming the flanges of the beam, Done
    Section: flange, OK
    # pick the region forming the web of the beam, Done
    Section: web, OK, X # to end the command
    # Create orientations to align the material accordingly
Menu:   Tools, Datum, Type: CSYS, Method: 2 lines
    Name [web-csys], Rectangular, Cont
    # pick a line along Z-global, pick a line along Y-global
    Name [flange-csys], Rectangular, Cont
    # pick a line along Z-global, pick a line along X-global
    # close the pop-up windows
Menu:   Assign, Material Orientation
    # pick the web, Done, Datum CSYS List, web-csys, OK,
        Normal: Axis 3, OK
    # pick the four regions forming the flange, Done
    Datum CSYS List, # pick flange-csys, OK, Normal: Axis 3, OK
```

iii. Creating the assembly

```
Module: Assembly
Menu:   Instance, Create, Independent, OK
```

iv. *Defining analysis steps*

```
Module: Step
Menu:   Step, Create, Cont, OK
```

v. *Create constraints*

```
Module: Interaction
    # create a reference point (RP) at the loaded end of the I-beam
Menu:   Tools, Reference Point, [0,0,0]
    # define rigid body constraint of the loaded end
    # Tie nodes makes all disp and rota equal to those of the RP
Menu:   Constraint, Create, Rigid Body, Cont
    Tie (nodes), Edit, # pick the 5 lines at the loaded end, Done
    Reference Point: Edit, # pick the RP created, OK
```

vi. *Adding loads and BC*

```
Module: Load
Menu:   Load, Create
    Step: Step-1, Mechanical, Concentrated force, Cont
    # select the RP, Done, CF3 [11452], OK
Menu:   BC, Create
    Step: Initial, Mechanical, Symm/Anti/Enca, Cont
    # pick lines opposite to loaded end, Done, ZSYMM, OK
```

vii. *Meshing the model*

```
Module: Mesh
Menu:   Seed, Edges
    # pick the six longitudinal lines, Done
    Method: By Number, Bias: None, Number of elements [10], Apply, OK
    # pick edges of the flanges at both ends of the beam, Done
    Method: By Number, Bias: None, Number of elements [3], Apply, OK
    # pick the edges of the web at both ends of the beam, Done
    Method: By Number, Bias: None, Number of elements [4], Apply, OK
Menu:   Mesh, Controls, # select all, Done, Technique: Structured, OK
Menu:   Mesh, Element Type, # select all, Done
   Geometric Order: Quadratic, DOF: 6, Family: Shell, OK # S8R in use
Menu:   Mesh, Instance, Yes
```

Create a set containing the central point of the end opposite to the loaded end. This point will be used to apply an extra BC to prevent the movement on the symmetry plane.

```
Module: Mesh
Menu:   Tools, Set, Create, Name [Cpoint], Type: Node, Cont
    # pick the point at the center of unloaded end, Done
Module: Load
Menu:   BC, Create
    Step: Initial, Mechanical, Disp/Rota, Cont
    Sets, Cpoint, Cont, # checkmark: U1,U2,UR3, OK
```

viii. Solving and visualizing the results

```
Module: Job
Menu:   Job, Manager
   Create, Cont, OK, Submit, # when completed, Results
Module: Visualization
Menu:   Plot, Allow Multiple Plot States
Menu:   Plot, Deformed Shape
   Toolbar: Field Output, Primary, U, U3
```

Notice that the displacement at the loaded end is 0.03584 *mm.*

3.3 Failure Criteria

Failure criteria are curve fits of experimental data that attempt to predict failure under multiaxial stress based on experimental data obtained under uniaxial stress. All failure criteria described in this section predict the first occurrence of failure in one of the laminas but are unable to track failure propagation until complete laminate failure. Damage mechanics is used in Chapters 8 and 9 to track damage evolution up to laminate failure. The truncated-maximum-strain criterion estimates laminate failure without tracking damage evolution by making certain approximations and assumptions about the behavior of the laminate [1].

In this section, failure criteria are presented using the notion of failure index, which is used for several FEA packages, and it is defined as

$$I_F = \frac{stress}{strength} \tag{3.19}$$

Failure is predicted when $I_F \geq 1$. The strength ratio ([1, Section 7.1.1]) is the inverse of the failure index

$$R = \frac{1}{I_F} = \frac{strength}{stress} \tag{3.20}$$

Failure is predicted when $R \leq 1$.

3.3.1 2D Failure Criteria

Strength-based failure criteria are commonly used in FEA to predict failure events in composite structures. Numerous criteria exist for unidirectional (UD) laminas subjected to a state of plane stress ($\sigma_3 = 0$). The most commonly used are described in [1]. They are:

- Maximum stress criterion
- Maximum strain criterion
- Truncated maximum strain criterion, and
- Interacting failure criterion

A few additional criteria are presented in this section.

Hashin Failure Criterion

The Hashin failure criterion (HFC) proposes four separate modes of failure:

- Fiber tension
- Fiber compression
- Matrix tension
- Matrix compression

that are predicted by four separate equations, as follows:[4]

$$I_{Fft}^2 = \left(\frac{\sigma_1}{F_{1t}}\right)^2 + \alpha\left(\frac{\sigma_6}{F_6}\right)^2 \qquad \text{if } \sigma_1 \geq 0 \tag{3.21}$$

$$I_{Ffc}^2 = \left(\frac{\sigma_1}{F_{1c}}\right)^2 \qquad \text{if } \sigma_1 < 0 \tag{3.22}$$

$$I_{Fmt}^2 = \left(\frac{\sigma_2}{F_{2t}}\right)^2 + \left(\frac{\sigma_6}{F_6}\right)^2 \qquad \text{if } \sigma_2 \geq 0 \tag{3.23}$$

$$I_{Fmc}^2 = \left(\frac{\sigma_2}{2F_4}\right)^2 + \left[\left(\frac{F_{2c}}{2F_4}\right)^2 - 1\right]\frac{\sigma_2}{F_{2c}} + \left(\frac{\sigma_6}{F_6}\right)^2 \qquad \text{if } \sigma_2 < 0 \tag{3.24}$$

where α is a weight factor to give more or less emphasis to the influence of shear on fiber failure. With $\alpha = 0$, Hashin FC and Maximum Stress FC would predict longitudinal tensile failure of the unidirectional lamina at the same stress σ_1. Note (3.22) predicts longitudinal compressive failure without influence of the shear stress, although it is well known that the state of shear has a strong influence on the longitudinal compression failure [21].

Equations (3.21–3.24) define the *square* of failure indexes according to HFC. *Abaqus* documentation refers to the square of failure indexes for HFC, Tsai-Hill, Azzi-Tsai-Hill, and Tsai-Wu[5], in contrast to the (*not squared*) failure indexes for maximum stress (MSTRS) and maximum strain (MSTRN) failure criteria.

The values of F_{1c} and F_{2c} are considered positive throughout this textbook and most of the literature, but in *Abaqus* they must be given with different signs in different areas of *Abaqus*. Specifically, when entering strength values in `Edit Material`, `Material Behavior: Elastic`, `Suboption: Fail Stress`, F_{1c} and F_{2c} must be entered as negative values. However, when entering the same values to define the threshold of damage initiation in `Edit Material`, `Material Behavior: Hashin Damage`, they must be given as positive quantities.

Although the threshold of damage initiation is calculated in *Abaqus* by the Hashin failure criterion, the subsequent analysis of damage accumulation is done by a decohesion methodology that is independent of the damage initiation criterion

[4] The Hashin failure equations on the right-hand side of (3.21–3.24) yield squares of failure indexes I_F. Compare to the Maximum Stress Criterion (i.e., $I_{Fft} = \sigma_1/F_{1T}$) and so on.

[5] The Tsai-Hill, Azzi-Tsai-Hill, and Tsai-Wu failure criteria are not recommended because they overemphasize the interaction between fiber (σ_1) and transverse matrix (σ_2) damage modes.

used. Despite this, *Abaqus* documentation and *Abaqus*/CAE refer to the whole damage methodology as *Hashin Damage*.

Puck Failure Criterion

The Puck failure criterion [22] distinguishes between fiber failure (FF) and matrix failure (MF). In the case of plane stress, the MF criteria discriminates three different modes. Mode A is when transverse cracks appear in the lamina under transverse tensile stress with or without in-plane shear stress. Mode B also denotes transverse cracks, but in this case they appear under in-plane shear stress with small transverse compression stress. Mode C indicates the onset of oblique cracks (typically with an angle of 53° in carbon epoxy laminates) when the material is under significant transverse compression.

The FF and the three MF modes yield separate failure indexes. The Puck criterion assumes that FF only depends on longitudinal tension. Therefore, the failure index for FF is defined as

$$I_{FF} = \begin{cases} \sigma_1/F_{1t} & \text{if } \sigma_1 > 0 \\ -\sigma_1/F_{1c} & \text{if } \sigma_1 < 0 \end{cases} \tag{3.25}$$

The MF failure indexes have different expressions depending on the mode that becomes active. With positive transverse stress, mode A is active. In this case, the failure index for matrix dominated tensile failure (mode A) is

$$I_{MF,A} = \sqrt{\left(\frac{\sigma_6}{F_6}\right)^2 + \left(1 - p_{6t}\frac{F_{2t}}{F_6}\right)^2 \left(\frac{\sigma_2}{F_{2t}}\right)^2} + p_{6t}\frac{\sigma_2}{F_6} \quad \text{if } \sigma_2 \geq 0 \tag{3.26}$$

where p_{6t} is a fitting parameter. Lacking experimental values, it is assumed that $p_{6t} = 0.3$ [22].

Under negative transverse stress, either mode B or mode C is active, depending on the relationship between in-plane shear stress and transverse shear stress. The limit between mode B and C is defined by the relation F_{2A}/F_{6A}, where

$$F_{2A} = \frac{F_6}{2p_{6c}}\left[\sqrt{1 + 2p_{6c}\frac{F_{2c}}{F_6}} - 1\right] \tag{3.27}$$

$$F_{6A} = F_6\sqrt{1 + 2p_{2c}} \tag{3.28}$$

and p_{2c} is defined as

$$p_{2c} = p_{6c}\frac{F_{2A}}{F_6} \tag{3.29}$$

and p_{6c} is another fitting parameter. Lacking experimental values, it is assumed that $p_{6c} = 0.2$ [22].

Finally, the failure index for matrix dominated shear (mode B) is

$$I_{MF,B} = \frac{1}{F_6}\left[\sqrt{\sigma_6^2 + (p_{6c}\sigma_2)^2} + p_{6c}\sigma_2\right] \quad \text{if } \begin{cases} \sigma_2 < 0 \\ \left|\frac{\sigma_2}{\sigma_6}\right| \leq \frac{F_{2A}}{F_{6A}} \end{cases} \tag{3.30}$$

and for matrix dominated compression (mode C) is

$$I_{MF,C} = \ -\frac{F_{2c}}{\sigma_2}\left[\left(\frac{\sigma_6}{2(1+p_{2c})F_6}\right)^2 + \left(\frac{\sigma_2}{F_{2c}}\right)^2\right] \ \text{if} \ \begin{cases} \sigma_2 < 0 \\ \left|\frac{\sigma_2}{\sigma_6}\right| \geq \frac{F_{2A}}{F_{6A}} \end{cases} \tag{3.31}$$

3.3.2 3D Failure Criteria

Failure criteria presented here are 3D generalizations of the ones presented in [1, Section 7.1]. The 2D version of the criteria presented in this section can be recovered by setting $\sigma_4 = \sigma_5 = 0$. The user of FEA packages should be careful because some packages use only the in-plane stress components for the computation of the failure index (e.g., *Abaqus*), even though all six stress components may be available from the analysis. In those cases the intralaminar and thickness components of stress should be evaluated separately to see if they lead to failure.

In this section, the numerical subscript denotes the directions of 1) fiber, 2) in-plane transverse to the fibers, and 3) through the thickness of the lamina. The letter subscript denotes t) tensile and c) compressive. Contracted notation is used for the shear components as described in Section 1.5.

Maximum Strain Criterion

The failure index is defined as

$$I_F = \max \begin{cases} \epsilon_1/\epsilon_{1t} & \text{if } \epsilon_1 > 0 \quad \text{or} \quad -\epsilon_1/\epsilon_{1c} \quad \text{if } \epsilon_1 < 0 \\ \epsilon_2/\epsilon_{2t} & \text{if } \epsilon_2 > 0 \quad \text{or} \quad -\epsilon_2/\epsilon_{2c} \quad \text{if } \epsilon_2 < 0 \\ \epsilon_3/\epsilon_{3t} & \text{if } \epsilon_3 > 0 \quad \text{or} \quad -\epsilon_3/\epsilon_{3c} \quad \text{if } \epsilon_3 < 0 \\ abs(\gamma_4)/\gamma_{4u} & \\ abs(\gamma_5)/\gamma_{5u} & \\ abs(\gamma_6)/\gamma_{6u} & \end{cases} \tag{3.32}$$

The quantities in the denominator are the ultimate strains of the unidirectional lamina. Note that compression ultimate strains in (3.32) are positive numbers.

Maximum Stress Criterion

The failure index is defined as

$$I_F = \max \begin{cases} \sigma_1/F_{1t} & \text{if } \sigma_1 > 0 \quad \text{or} \quad -\sigma_1/F_{1c} \quad \text{if } \sigma_1 < 0 \\ \sigma_2/F_{2t} & \text{if } \sigma_2 > 0 \quad \text{or} \quad -\sigma_2/F_{2c} \quad \text{if } \sigma_2 < 0 \\ \sigma_3/F_{3t} & \text{if } \sigma_3 > 0 \quad \text{or} \quad -\sigma_3/F_{3c} \quad \text{if } \sigma_3 < 0 \\ abs(\sigma_4)/F_4 & \\ abs(\sigma_5)/F_5 & \\ abs(\sigma_6)/F_6 & \end{cases} \tag{3.33}$$

The letter F is used here to denote a strength value for a unidirectional lamina as in [23]. Note that compression strength in (3.33) are positive numbers.

Tsai-Wu Criterion

Using the Tsai-Wu criterion the failure index is defined as

$$I_F = \frac{1}{R} = \left[-\frac{B}{2A} + \sqrt{\left(\frac{B}{2A}\right)^2 + \frac{1}{A}} \right]^{-1} \tag{3.34}$$

with

$$\begin{aligned} A = & \frac{\sigma_1^2}{F_{1t}F_{1c}} + \frac{\sigma_2^2}{F_{2t}F_{2c}} + \frac{\sigma_3^2}{F_{3t}F_{3c}} + \frac{\sigma_4^2}{F_4^2} + \frac{\sigma_5^2}{F_5^2} + \frac{\sigma_6^2}{F_6^2} \\ & + c_4\frac{\sigma_2\sigma_3}{\sqrt{F_{2t}F_{2c}F_{3t}F_{3c}}} + c_5\frac{\sigma_1\sigma_3}{\sqrt{F_{1t}F_{1c}F_{3t}F_{3c}}} + c_6\frac{\sigma_1\sigma_2}{\sqrt{F_{1t}F_{1c}F_{2t}F_{2c}}} \end{aligned} \tag{3.35}$$

and

$$B = \left(F_{1t}^{-1} - F_{1c}^{-1}\right)\sigma_1 + \left(F_{2t}^{-1} - F_{2c}^{-1}\right)\sigma_2 + \left(F_{3t}^{-1} - F_{3c}^{-1}\right)\sigma_3 \tag{3.36}$$

where c_i, $i = 4..6$, are the Tsai-Wu coupling coefficients that by default are taken to be -1. Note that compression strength in (3.35) and (3.36) are here positive numbers.

The through-the-thickness strength values F_{3t} and F_{3c} are seldom available in the open literature, so it is common practice to use the corresponding in-plane transverse values of strength. Also, the intralaminar strength F_5 is commonly assumed to be equal to the in-plane shear strength. The intralaminar strength F_4 can be estimated [1, (4.109)] in terms of transverse compression strength F_{2c} and the angle of the fracture plane α_0. The latter is consistently reported to be close to 53^o.

Example 3.12 *Compute the failure index I_F and strength ratio R in each layer of Example 3.4 (p. 121) using the Maximum Stress Failure criterion and the Tsai-Wu criterion. The elastic moduli and strength values for the lamina are given in Table 3.1, p. 102. Determine the strength ratio of the laminate using both criteria.*

Solution to Example 3.12 *Hashin, Maximum Strain, and Maximum Stress failure criteria are adequate failure criteria in* Abaqus, *but Tsai-Hill and Tsai-Wu failure criteria are not recommended because they overestimate the interaction between σ_1 (S11) and σ_2 (S22). Tsai-Wu is used in this example to be consistent with [8, Example 3.12].*

Unlike the definition used in this textbook (3.34), Abaqus *defines the Tsai-Wu failure index as:*

$$TSAIW = f1 * S11 + f2 * S22 + f11 * S11^2 + f22 * S22^2 + f66 * S12^2 + 2f12 * S11 * S22$$

which is a square *of a failure index. Therefore, the values of TSAIW do not compare with the failure indices for MSTRN* (3.32) *and MSTRS* (3.33) *unless a square root of TSAIW is taken. The same happens for the Tsai-Hill failure index TSAIH and for the four Hashin failure indexes in* Abaqus *(compare the* Abaqus *documentation to (3.21–3.24)).*

The strength values F1T, F1C, F2T, F2C, F6, f12, and `Slim` *are supplied to the* Abaqus*/CAE using* `Edit Material`, `Material Behavior: Elastic`, `Suboption: Fail Stress`. *Note that F1C and F2C must be given as negative numbers for TSAIW but as* *positive numbers for Hashin Damage (see Section 3.3.1). Also,* `f12` *and the biaxial stress* `Slim` *default to zero if not given, which is good, because interaction* `f12` *overestimates the interaction between σ_1 (S11) and σ_2 (S22). However, in this example, we use $f12 = -1$ to reproduce the results of [8, Example 3.12].*

To solve this example, open `Ex_3.4.a.cae` *and Save As* `Ex_3.12.cae`*. Then, do the following:*

i. Set `Work Directory`

```
Menu:   File, Set Work Directory, [C:\SIMULIA\User\Ex_3.12], OK
```

ii. Defining the damage parameters for the material

```
Module: Property
Menu:   Material, Manager
    Edit, Suboptions, Fail Stress, # you can copy/paste from excel
    [1830    -1096    57    -228    71    -1    0], OK, OK
    # close Material Manager pop-up window
```

iii. Defining analysis step to create a field output for the failure index

```
Module: Step
Step:   Step-1
Menu:   Output, Field Output Requests, Edit, F-Output-1
    # expand: Failure/Fracture, # checkmark: CFAILURE
    # uncheck: everything else we don't need, OK
```

iv. Solving and visualizing the results

```
Module: Jobs
Menu:   Job, Manager
    Submit, # when completed, Results
Module: Visualization
Menu:   Plot, Contours, On Deformed Shape
Menu:   Result, Field Output, MSTRS, OK
Menu:   Results, Section Points, Method: Plies
    # pick a ply, Apply, # read the value, OK
Menu:   Result, Field Output, TSAIW, OK
Menu:   Results, Section Points, Method: Plies
    # pick a ply, Apply, # read the value, OK
```

The results are tabulated in Table 3.7. You should compare the results to the classical lamination theory (CLT) solution using [12].

Example 3.13 *[8, Example 3.14] Compute the Tsai-Wu failure index I_F on each layer of Example 3.8 (p. 151) using a UMAT subroutine, available in [5,* `umatS8R5.for`*]. The elastic moduli and strength values for the lamina are given in Table 3.1, p. 102.*

Solution to Example 3.13 *Environment variables must be set properly before executing a job with user programmable features (see Appendix C). To solve this example, first open* `Ex_3.8.cae` *and save it as* `Ex_3.14.cae`*. Also, set the Work Directory to the local directory so that all the files are easily accessed.*

Table 3.7: Index failure MSTRS (Maximum Stress Criterion), and TSAIW (Tsai-Wu) calculated in Example 3.12.

Ply #	MSTRS		TSAIW	
	I_F	R	I_F	R
1	0.01442	69.34813	0.01447	69.10850
2	0.02430	41.15226	0.03107	32.18539
3	0.01566	63.85696	0.01873	53.39028
4	0.01566	63.85696	0.01873	53.39028

```
Menu:   File, Set Work Directory, [C:\SIMULIA\User\Ex_3.14], OK
Menu:   File, Open, [C:\SIMULIA\User\Ex_3.8\Ex_3.8.cae], OK
Menu:   File, Save As, [C:\SIMULIA\User\Ex_3.14\Ex_3.14.cae], OK
```

Only a few things need to be modified in the .mdb. Remember that in Abaqus *the stress/strain components for shell elements are ordered as follows: 11, 22, 12, 13, 23. Then, the* section shear stiffness *can be calculated using [12]. Alternatively, for a single ply shell of thickness t, use*

$$H_{44} = (5/6)\ t\ G_{23}$$
$$H_{45} = 0$$
$$H_{55} = (5/6)\ t\ G_{13}$$

or for a laminate use [1, (6.16)], or use Abaqus *to calculate the values and read them from the* `.dat` *file as illustrated below. To calculate with* Abaqus, *start with the* `.cae` *of Example 3.8 (p. 151) and modify it as follows:*

i. Modifying the material definition to add failure parameters

```
Module: Property
# unidirectional material from Ex. 3.8 used to calculate H44, H45, H55
Menu:   Material, Create
    Name [unidirectional], Mechanical,
    Elasticity, Elastic, Type: Lamina
    [133860     7706     0.301     4306     4306     2760]
    # or right-click on Data field, Read from File
    Select, [uni_lam_prop.txt], OK, OK
    # add Fail Stress values
    Suboptions, Fail Stress
    [1830    -1096    57    -228    71    -1], OK
    # or right-click on Data field, Read from File
    Select, [uni_fail_stress.txt], OK, OK, OK,
    OK # to close Edit Material
Menu:   Section, Edit, Section-1
    Tab: Basic, Material: unidirectional, OK
```

ii. Submitting the job and visualizing the results

```
Module: Job
Menu:   Job, Manager
```

```
        Edit, Tab: General, # checkmark: Print model definition data, OK
        Submit, # when completed, Results
        Field Output Toolbar: Primary, U, U3, # U3max = 8.704 mm
```

Use a text editor to open the file `Job-1.dat`*. The file must be stored in the* ***Work Directory*** *(*`C:\SIMULIA\User\Ex_3.14`*). Read the values corresponding to the Transverse Shear Stiffness for the Section. They should be:*

$$Kts11 = K11 = H55 = 35883.0$$
$$Kts22 = K22 = H44 = 23000.0$$
$$Kts12 = K12 = H45 = 0.0$$

To run the user material subroutine, do the following:

i. *Create a user material to be used with the UMAT subroutine and edit the corresponding section*

```
Module: Property
Menu:   Material, Create
    Name [ud-user], General, User Material,
    User material type: Mechanical
    Mechanical constants, [133860   7706   0.301   0.396
    4306   1830   1096   57   228   71   -1]
    # or right-click on Data field, Read from File
    Select, [user_mat_props.txt], OK, OK
    # see umats8r5.for for interpretation of Mechanical constants
    General, Depvar,
    Number of solution-dependent state variables [2], OK
Menu:   Section, Edit, Section-1
    Tab: Basic
    Section integration: During analysis
    Thickness: Shell thickness: Value [10]
    Material: ud-user
    Thickness integration rule: Simpson
    Thickness integration points [3]
    Tab: Advanced
    Transverse Shear Stiffness: # checkmark: Specify values
    K11: [35883], K12: [0], K22: [23000], OK
```

ii. *Define output variables for the analysis step*

```
Module: Step
Step:   Step-1
Menu:   Output, Field Output Requests, Edit, F-Output-1
    Output variables: Edit variables: [S,E,U,RF,SDV], OK
```

iii. *Submit the job and visualize the results.*
Before submitting the job make sure your system is properly setup to run a UMAT. Notice that if your system is not properly setup to run UMATs you will get an error after submitting the job for execution.

```
Module: Job
Menu:   Job, Manager
```

```
        Edit, Tab: General,
        User subroutine file: Select, [umatS8R5.for], OK, OK
        Submit, # when completed, Results
    Module: Visualization
    Menu:   Options, Contour, Contour Type: Quilt, OK
    Menu:   Plot, Contours on Deformed Shape
        Field Output Toolbar: Primary, SDV1
    Menu:   Result, Section Points, Selection method: Plies
        Ply result location: Bottommost, Apply
        Ply result location: Topmost, Apply
        OK # close the Section Points pop-up window
```

In case of an error, check that the configuration file `C:\SIMULIA\Abaqus\Commands\abq6102.bat` *is correct. Also, check that the following is included in the* `.inp` *file:*

```
** Section: Section-1
*Shell Section, elset=_PickedSet2, material=uni-user, orientation=Ori-1
10., 3
*Transverse Shear
35883., 23000., 0.
```

and

```
*Material, name=uni-user
*Depvar
2,
*User Material, constants=11
133860., 7706., 0.301, 0.396, 4306., 1830., 1096.,   57.
228.,   71.,   -1.
```

From the visualization of `SDV1` *on a quilted contour, the bottom side reaches* $I_F = 1.767 \times 10^{-2}$ *and the top side reaches* $I_F = 9.444 \times 10^{-3}$.

3.4 Predefined Fields

The following example illustrates the analysis of thermal expansion and how to introduce a `Predefined Field` in general.

Example 3.14 *Display the deformed shape of a* $[0/90]_T$ *laminate illustrating thermal expansion coupling (discussed in [1, Section 6.1.4, Figure 6.7]). Each lamina is* 1.2 *mm thick, made of AS4D/9310 (Table 3.1). The geometry of the plate is square with* $a_x = a_y = 1000\,mm$. *The thermal loading is* $\Delta\, T = -125^\circ\, C$. *Use element S8R5 in* Abaqus.

Solution to Example 3.14 *A* $[0/90]_T$ *laminate under uniform thermal loading* $\Delta\, T$, *with square geometry* ($a_x = a_y$) *on the x-y plane, is symmetric with respect to the x-z and y-z planes, so only 1/4 of the plate needs to be modeled. Note that the fact that the LLS is not symmetric does not affect the symmetry of the discretization, which is determined by three factors: (a) symmetry of geometry, (b) symmetry of load, and (c) symmetry of material, the latter on the x-y plane only. The following pseudo code illustrates the procedure. The saddle shape of the deformed plate is shown in Figure 3.22. The CAE file is available in [5, Ex_3_15.cae].*

i. *Geometry*

```
Module: Part
Menu:   Part, Create, [Part-1], 3D, Deformable,
    Shell, Planar, Approximate size [4000], Cont
Menu:   Add, Line, Rectangle,
    # Enter the points [0,0], [1000,1000], X, Done
```

ii. *Material and Section Properties*

```
Module: Property
Menu:   Material, Create,
    Name [Material-1], Mechanical, Elasticity, Elastic, Lamina,
    [13386000 7706 0.301 4306 4306 2760]
    Mechanical, Expansion, Type, Orthotropic,
    [13.23672E-007 2.58908E-005 2.58908E-0051], OK
Menu:   Section, Create, Name [Section-1], Shell, Composite, Cont
    Material: Material-1 # for all
    Thickness [1.2] # for all
    Orientation [0,90]
Menu:   Assign, Section
    # pick part, Done, OK
```

iii. *Assembly and Step*

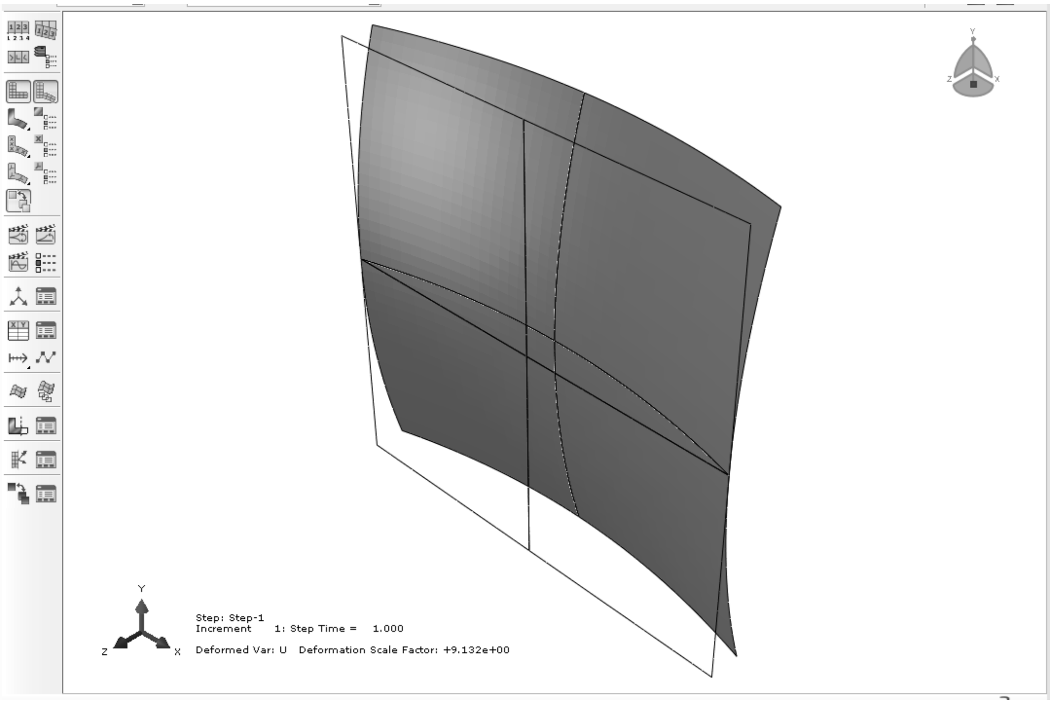
```
Module: Assembly
Menu:   Instance, Independent, OK

Module: Step
Menu:   Step, Create, Name [Step-1], Static/General, Cont, OK
```

iv. *Load and Boundary Conditions*

```
Module: Load
Menu:   BC, Create, Name [BC-1], Step: Initial,
    Mechanical, Symmetry, Cont
    # pick left vertical line, Done, XSIMM, OK
Menu:   BC, Create, Name [BC-2], Step: Initial,
    Mechanical, Symmetry, Cont
    # pick lower horizontal line, Done, YSIMM, OK
Menu: BC, Create, Displacement/Mechanical, Cont,
    # pick bottom left point, Done
    # checkmark all (U1 to UR3), OK
Menu:   Predefined Field, Create, Other, Temperature, Cont
    # Select rectangle, Done, Magnitude [0], OK
Menu:   Predefined Field, Manager
    # Double click cell Predefined Field-2, Step-1 (Propagated)
    Status, Modified, Magnitude [-150], OK
```

v. *Mesh*

```
Module: Mesh
Menu:   Seed, Instance, Apply, OK
Menu:   Mesh, Element type, Std, Quad, DOF [5], OK # S8R5
Menu:   Mesh, Instance, Yes
```



```
Step: Step-1
Increment     1: Step Time =    1.000
Deformed Var: U   Deformation Scale Factor: +9.132e+00
```

Fig. 3.22: Deformed shape in Example 3.14 under uniform thermal load.

vi. Job and Visualization

```
Module: Jobs
Menu:   Job manager
    Create, Cont, OK
    Submit,
    Results

Module: Visualization
Menu:   Plot,
    Allow multiple plot states, Undeformed shape, Deformed shape
Menu:   Options, Common,
    Tab: Basic, Render style: Shaded, Visible Edges: Free edges,
    Tab: Other, Scale coordinates, Z: [3.0], OK
Menu:   Options, Superimpose plot options,
    Visible Edges: Feature edges, OK
Menu:   View, Odb display options,
    Tab: Mirror/Pattern, Mirror planes: XZ and YZ
```

Suggested Problems

Problem 3.1 *Compute the maximum bending moment per unit cross-sectional area m_u that can be applied to a beam of circular hollow cross section of outside radius r_o and* *inner radius r_i. The loading is pure bending, no shear. The material is homogeneous and failure occurs when the maximum stress reaches the strength σ_u of the material. The hollow section is filled with foam to prevent buckling. Derive an expression for the efficiency of the cross section as the ratio of m_u of the hollow beam by m_u of a solid rod of same outside radius. Faced with the problem of using a strong and relatively expensive material, would you recommend a small or large radius?*

Problem 3.2 *Compute the maximum outside radius for a cantilever beam of length L, loaded by a tip load P, otherwise similar to the beam in Problem 3.1 but subjected to pure shear loading. The shear strength is $\tau_u = \sigma_u/2$. Consider only shear. Buckling of the thin wall is likely to limit further the practical thickness of the wall.*

Problem 3.3 *Compute the maximum deflection per unit volume δ_V that can be applied to a beam of circular hollow cross section of outside radius r_o and inner radius r_i. This is a cantilever beam of length L, loaded by a tip load P. The hollow section is made of an homogeneous material with moduli E and $G = E/2.5$, filled with foam to prevent buckling. Derive an expression for the efficiency of the cross section as the ratio of δ_V between the hollow cross section and a solid rod of the same outside radius. Faced with the problem of using a relatively expensive and not quite stiff material, would you recommend a small or large radius?*

Problem 3.4 *Write a computer program to evaluate (3.9). The program data input is the LSS, the thickness of the laminas, and the material elastic properties. The output should be written in a file. Show all work in a report.*

Problem 3.5 *Using the program of Problem 3.4 compute the A, B, D, and H matrices for the following laminates. The material is AS4D/9310 and all layers are* 0.85 *mm thick. Comment on the coupling of the constitutive equations for each case: (a) one layer* $[0]$, *(b) one layer* $[30]$, *(c)* $[0/90]_2$, *(d)* $[0/90]_s$, *(e)* $[0/90]_8$, *(f)* $[\pm 45]_2 = [+45/-45/+45/-45]$, *(g)* $[\pm 45]_s = [+45/-45/-45/+45]$, *(h)* $[\pm 45/0/90/\pm 30]$. *Show all work in a report.*

Problem 3.6 *Compute the value and location of the absolute maximum transverse shear strain γ_{23} in Example 3.2, p. 113. At that location, plot the distribution of γ_{23} through the thickness of the plate. Is that distribution a reasonable answer?*

Problem 3.7 *Recompute Example 3.2, p. 113, with a doubly sinusoidal load*

$$q(x, y) = q_0 \sin(\pi x/2a) \sin(\pi x/2b)$$

where $2a, 2b$ are the plate dimensions in x and y, respectively. Compare the result with the exact solution at the center of the plate, that is

$$w_0 = 16 q_0 b^4 / [\pi^4 (D_{11} s^4 + 2(D_{12} + 2D_{66}) s^2 + D_{22})]$$

where $s = b/a$ ([14, (5.2.8–5.2.10)]).

Problem 3.8 *Calculate the first vibration frequency ϖ_{11} of the plate with the analytical solution $\varpi_{mn}^2 = \pi^4 [D_{11} m^4 s^4 + 2(D_{12} + 2D_{66}) m^2 n^2 s^2 + D_{22} n^4]/(16 \rho h b^4)$, where ρ, h are the density and thickness of the plate, respectively ([14, (5.7.8)]).*

Problem 3.9 *Consider a rectangular plate with* $a_x = 1000$ *mm and* $b_y = 100$ *mm. Each lamina is* 1.2 *mm thick, made of AS4D/9310 (Table 3.1).*

Propose three different LSS, each having one of the following couplings (shown in [1, Figure 6.7]); that is: (a) bending-extension coupling, (b) torsion-extension coupling, and (c) shear-extension coupling.

Next, consider reduced FEA models applying symmetry conditions; either (i) one-half plate, 500×100 *mm, or (ii) one-quarter plate,* 500×50 *mm. Use the smallest (1/4, 1/2, or full plate) FEA model that is valid for each of the cases (a) through (c). Apply membrane loading* $N_x = 1000$ *N/mm (all other zero). Show all work in a report including pseudo code for each case (a) through (c). Use element S8R5 in* Abaqus. *See Example 3.14, p. 171.*

Problem 3.10 *A failure envelope is a 3D surface for which* $I_f = 1$. *The intersection of this surface with each coordinate plane is a 2D curve. Using a program such as* MATLAB, *plot the failure envelope in the plane* $\sigma_1 - \sigma_2$ *for Maximum Stress, Tsai-Wu, and Puck failure criteria in the same plot. Do a similar plot in the plane* $\sigma_2 - \sigma_6$.

Problem 3.11 *(a) Compute the failure index* I_F *on each layer of Example 3.12 (p. 167) using the* Maximum Stress Failure Criterion *and the* Hashin failure criterion. *The lamina strength values are given in Table 3.1.*

Problem 3.12 *Using UMAT subroutine, compute the failure index* I_F *on each layer of Example 3.12 (p. 167) using the Puck failure criterion. The elastic moduli and strength values for the lamina are given in Table 3.1, p. 102. Values for the Puck parameters are suggested in Section 3.3.1, p. 165. Show all work in a report.*

Problem 3.13 *Write on a text file the nodal stress results at the top and bottom of each lamina of Problem 3.11. Then, using an external program such as* MATLAB, *compute the same failure indexes as in Problem 3.11 in terms of the nodal stress values at the center of the plate, and compare them to the results of Problem 3.11. Show all work in a report.*

Chapter 4

Buckling

Most composite structures are thin walled. This is a natural consequence of the following facts:

- Composites are stronger than conventional materials. Then, it is possible to carry very high loads with a small area, and thus small thickness in most components.

- Composites are expensive when compared to conventional materials. Therefore, there is strong motivation to reduce the volume, and thus the thickness as much as possible.

- The cost of polymer matrix composites increases with their stiffness. The stiffness in the fiber direction can be estimated by using the fiber dominated rule of mixtures, $E_1 = E_f V_f$. For example, when glass fibers are combined with a polymer matrix, the resulting composite stiffness is lower than that of aluminum. Using Aramid yields a stiffness comparable to aluminum. Carbon fibers yield composite stiffness lower than steel. Therefore, there is strong motivation to increase the moment of inertia of beams and stiffeners without increasing the cross-sectional area. The best option is to increase the moment of inertia by enlarging the cross section dimensions and reducing the thickness.

All the above factors often lead to design of composite structures with larger, thin walled cross sections, with modes of failure likely to be controlled by buckling.

4.1 Eigenvalue Buckling Analysis

Buckling is loss of stability due to geometric effects rather than material failure. But it can lead to material failure and collapse if the ensuing deformations are not restrained. Most structures can operate in a linear elastic range. That is, they return to the undeformed configuration upon removal of the load. Permanent deformations result if the elastic range is exceeded, as when matrix cracking occurs in a composite.

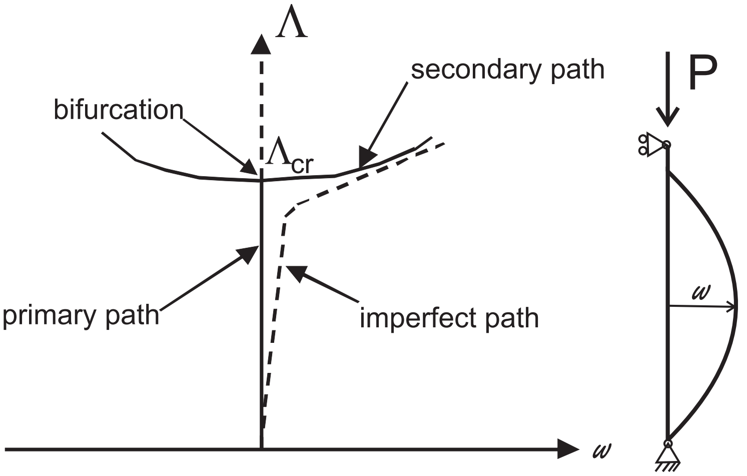


Fig. 4.1: Equilibrium paths for the perfect column.

Consider a simply supported column of area A, length L, and moment of inertia I, made of homogeneous material with modulus E and strength F along the length of the column. The column is loaded by a compressive load P acting on the centroid of the cross section [3]. If the column geometry, loading, and material have no imperfections, the axial deformation is

$$u = PL/EA \tag{4.1}$$

with no lateral deformation $w = 0$. The deformation of the structure (u, v, w) before buckling occurs is called the *primary path* (Figure 4.1). The slightest imperfection will make the column buckle when

$$P_{cr} = \pi^2(EI)/L^2 \tag{4.2}$$

The load capacity for long slender columns will be controlled by buckling, as opposed to the crushing strength of the material. What happens after the column reaches its critical load depends largely on the support conditions. For the simply supported column, the lateral deflection[1]

$$w = A\sin(\pi x/L) \tag{4.3}$$

will grow indefinitely ($A \to \infty$) when the load just barely exceeds P_{CR}. Such large lateral deflections will cause the material to fail and the column will collapse. The behavior of the structure after buckling has occurred is called *post-buckling*.

The simply supported column in Figure 4.1 experiences no deformations in the shape of the buckling mode (4.3) before buckling actually happens. In this case, it is said that the structure has a *trivial* primary path. This is a consequence of having a perfect structure with perfectly aligned loading. For these type of structures,

[1] With x measured from one end of the column.

Table 4.1: Lamina elastic properties for AS4/9310, $V_f = 0.6$.

Young's Moduli	Shear Moduli	Poisson's Ratio
$E_1 = 145880$ MPa	$G_{12} = G_{13} = 4386$ MPa	$\nu_{12} = \nu_{13} = 0.263$
$E_2 = E_3 = 13312$ MPa	$G_{23} = 4529$ MPa	$\nu_{23} = 0.470$

buckling occurs at a bifurcation point. A bifurcation point is the intersection of the primary path with the secondary path, i.e., the post-buckling path [24].

The bifurcation loads, one for every possible mode of buckling, are fairly easy to obtain using commercial software. The geometry of the structure is that of the perfect undeformed configuration, loaded with the nominal loads, and the material is elastic. Such analysis requires a minimum of effort on the part of the analyst. Commercial programs refer to this analysis as an *eigenvalue buckling analysis* because the critical loads are the eigenvalues λ_i of the discretized system of equations

$$([K] - \lambda[K_s])\{v\} = 0 \tag{4.4}$$

where K and K_s are the stiffness and stress stiffness matrix, respectively, and v is the column of eigenvectors (buckling modes) [24].

Example 4.1 *Consider a simple supported plate, with side dimensions $a_x = 1000$ mm, $a_y = 500$ mm, edge-wise loaded in compression with $N_x = N_y = 1$ N/mm. The plate is made of $[(0/90)_3]_S$, AS4/9310 (Table 4.1) with fiber volume fraction 0.6 and total thickness $t_T = 10.2$ mm. Compute the critical load of the lowest four modes using eigenvalue analysis. Visualize the lower four modes.*

Solution to Example 4.1 *We model the whole plate to capture all buckling modes, not just the symmetric ones. Element S8R5 is used because the we are modeling a thin plate. Since the laminate is symmetric and stress computation lamina by lamina is not required, the critical loads can be obtained using three different approaches. Note that for element S8R5, you are not required to specify `Transverse Shear Stiffness` when the shell section is `Homogeneous` or `Composite` (see cases (a) and (c)), but they have to be entered for a `General Stiffness` section (see case (b)).*

(a) Using Equivalent Laminate Moduli

In this first approach, the equivalent laminate moduli are calculated and used along with an orthotropic shell element. In this case, laminate moduli represent the stiffness of an equivalent orthotropic plate that behaves like the actual laminate under in-plane loads (see Section 3.2.10). Laminate moduli can be found as explained in Section 1.15. Introduce the lamina properties (Table 4.1) into (1.91), rotate each layer (1.53), add then according to (1.102) get the laminate moduli (1.105) listed in Table 4.2. Bending moduli [1, (6.36)] may give better results for this example.

The following pseudo code illustrates the modeling, solution, and visualization procedure using the GUI in Abaqus/*CAE. The corresponding* `Ex_4.1.a.cae` *file is available on the Web site [5]. The results are tabulated in Table 4.3.*

```
# Ex.4.1.a using equivalent laminate moduli
Menu:   File, Set Work Directory [C:\SIMULIA\user\Ex_4.1], OK
Menu:   File, Save As [C:\SIMULIA\user\Ex_4.1\Ex_4.1.a.cae], OK
```

Table 4.2: Equivalent laminate moduli for $[(0/90)_3]_S$.

Young's Moduli	**Shear Moduli**	**Poisson's Ratio**
$E_x = 79985$ MPa	$G_{xy} = 4386$ MPa	$\nu_{xy} = 0.044$
$E_y = 79985$ MPa	$G_{yz} = 4458$ MPa	$\nu_{yz} = 0.415$
$E_z = 16128$ MPa	$G_{xz} = 4458$ MPa	$\nu_{xz} = 0.415$

i. *Creating the part*

```
Module: Part
Menu:   Part, Create
    3D, Deformable, Shell, Planar, Approx size [2000], Cont
Menu:   Add, Line, Rectangle, [-500,-250], [500,250]
    X # to close command, Done
```

ii. *Defining materials, sections, and assigning sections to parts*

```
Module: Property
Menu:   Material, Create
    Name [Material-1], Mechanical, Elasticity, Elastic
    Type: Engineering Constants
    [79985, 79985, 16128, 0.044, 0.415, 0.415, 4386, 4458, 4458], OK
Menu:   Section, Create
    Name [Section-1], Shell, Homogeneous, Cont
    Shell thickness: Value [10.2], Material: Material-1, OK
Menu:   Assign, Section
    # pick the part, Done, OK
```

iii. *Creating assembly*

```
Module: Assembly
Menu:   Instance, Create, Independent, OK
```

iv. *Defining the analysis steps*

```
Module: Step
Menu:   Step, Create
    Procedure type: Liner Perturbation, Buckle, Cont
    Number of eigenvalues requested [10], Vectors/iteration [10], OK
```

v. *Adding BC and loads*

```
Module: Load
Menu:   Load, Create
    Step: Step-1, Mechanical, Shell Edge Load, Cont
    # pick: edges @ x=500 & y=250, Done, Magnitude [1.0], OK
Menu: BC, Manager
    Create, Step: Initial, Mechanical, Disp/Rota, Cont
    # pick: the 4 edges, Done, # checkmark: U3, OK
    Create, Step: Initial, Mechanical, Disp/Rota, Cont
    # pick: edge @ x=-500, Done, # checkmark: U1, OK
    Create, Step: Initial, Mechanical, Disp/Rota, Cont
    # pick: edge @ y=-250, Done, # checkmark: U2, OK, # close Manager
```

vi. Meshing the model

```
Module: Mesh
Menu:   Seed, Instance, Size: [100], OK
Menu:   Mesh, Controls, Technique: Structured, OK
Menu:   Mesh, Element Type, Geometric Order: Quadratic, DOF: 5, OK
Menu:   Mesh, Instance, Yes
```

vii. Solving and visualizing the results

```
Module: Job
Menu:   Job, Manager
    Create, Name [Ex-4-1-a], Cont, OK, Submit, # when completed,
    Results
Module: Visualization
Menu:   Plot, Deformed Shape # buckling mode 1 will be displayed
Menu:   Result, Step/Frame, # pick a mode, Apply # to visualize
    # eigenvalues are shown in the WS and recorded in Ex-4-1-a.dat
```

(b) Using** $A-B-D-H$ **Matrices

In this second approach, $A-B-D-H$ matrices are used. To get the laminate properties (A, B, D, and H matrices), introduce the lamina properties (Table 4.1) into (3.9). The resulting laminate matrices are:

$$\begin{bmatrix} A & B \\ B & D \end{bmatrix} = \begin{bmatrix} 817036 & 35937.6 & 0 & 0 & 0 & 0 \\ 35937.6 & 817036 & 0 & 0 & 0 & 0 \\ 0 & 0 & 44737.2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 8.55845\ 10^6 & 311579 & 0 \\ 0 & 0 & 0 & 311579 & 5.60896\ 10^6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 387872 \end{bmatrix}$$

$$[H] = \begin{bmatrix} 37812.8 & 0 \\ 0 & 37964.7 \end{bmatrix}$$

The following pseudo code illustrates how to modify `Ex_4.1.a.cae` *to use the A–B–D–H matrices. The corresponding file is named* `Ex_4.1.b.cae` *and it is available on the Web site [5]. The results are tabulated in Table 4.3.*

```
# Ex.4.1.b using ABDH matrices
Menu:   File, Set Work Directory [C:\SIMULIA\user\Ex_4.1], OK
Menu:   File, Open [C:\SIMULIA\user\Ex_4.1\Ex_4.1.a.cae], OK
Menu:   File, Save As [C:\SIMULIA\user\Ex_4.1\Ex_4.1.b.cae], OK
```

i. Defining materials, sections, and assigning sections to parts

```
Module: Property
Menu:   Material, Delete, Material-1, Yes
Menu:   Section, Delete, Section-1, Yes
Menu:   Section, Create, Shell, General Shell Stiffness, Cont
    Tab: Stiffness
    [817036  35937.6        0       0       0       0
              817036        0       0       0       0
                      44737.2       0       0       0
```

```
                           8.56e6  311579       0
                                   5.61e6       0
                                           387872]
Tab: Advanced
Transverse Shear Stiffnesses: # checkmark: Specify values
K11 [37812.8], K12 [0], K22 [37964.7], OK
```

ii. *Defining analysis step*

```
Module: Step
Menu:   Step, Edit, Step-1
    Maximum number of interactions [70], OK
    # solve the problem using a smaller number for this quantity
    # the following error message may be displayed
    # TOO MANY ITERATIONS NEEDED TO SOLVE THE EIGENVALUE PROBLEM
```

iii. *Solving and visualizing the results*

```
Module: Job
Menu:   Job, Manager
    Create, Name [Ex-4-1-b], Cont, OK, Submit, # when completed,
    Results
    # visualize the results as it is indicated in part (a)
```

(c) Using the LSS

In this third approach, the laminate stacking sequence (LSS) and the lamina properties (Table 4.1) are entered. The following pseudo code illustrates how to change Ex_4.1.a.cae *to solve this part of the problem. The corresponding* Ex_4.1.c.cae *file is available on the Web site [5]. The results are tabulated in Table 4.3.*

```
# Ex_4.1.c using LSS
Menu:   File, Set Work Directory [C:\SIMULIA\user\Ex_4.1], OK
Menu:   File, Open [C:\SIMULIA\user\Ex_4.1\Ex_4.1.a.cae], OK
Menu:   File, Save As [C:\SIMULIA\user\Ex_4.1\Ex_4.1.c.cae], OK
```

i. *Defining materials, sections, and assigning sections to parts*

```
Module: Property
Menu:   Material, Delete, Material-1, Yes
Menu:   Section, Delete, Section-1, Yes
Menu:   Material, Create
    Name [Material-1], Mechanical, Elasticity, Elastic, Type: Lamina
    [1.4588e5, 1.3312e4, 2.63e-1, 4.386e3, 4.386e3, 4.529e3], OK
Menu:   Section, Create
    Shell, Composite, Cont
    # checkmark: Symmetric layers
    # right-click: Insert Row After, # until there are six rows
    Material: Material-1 # for all layers, Thickness [0.85]
    Orientation Angle: [0,90,0,90,0,90], Integration Points [3]
    Ply Name [k1, k2, k3, k4, k5, k6], OK
```

ii. *Defining analysis step*

Table 4.3: Bifurcation loads [N/mm].

Mode #	(a)	(b)	(c)
1	253.27	210.1	210.04
2	320.05	319.87	319.75
3	573.36	643.18	642.81
4	998.64	831.65	831.11

```
Module: Step
Menu:   Step, Edit, Step-1
    Maximum number of interactions [70]
    # solve the problem using a smaller number for this quantity
    # the following error message may be displayed
    # TOO MANY ITERATIONS NEEDED TO SOLVE THE EIGENVALUE PROBLEM
```

iii. Solving and visualizing the results

```
Module: Job
Menu:   Job, Manager
    Create, Name [Ex-4-1-c], Cont, OK, Submit,
    # when Completed, Results
    # visualize the results as it is indicated in part (a)
```

4.1.1 Imperfection Sensitivity

To illustrate the influence of imperfections in buckling, let us consider the solid lines in Figure 4.1. The lateral deflection is zero for any load below the bifurcation load P_{CR} that is on the primary path of the perfect structure. The primary path intersects the secondary path at the bifurcation point, for which the load is P_{CR}. The post-critical behavior of the column is indifferent and slightly stable. Indifferent means that the column can deflect right or left. Stable post-critical path means that the column can take a slightly higher load once it has buckled. For a column, this stiffening behavior is so small that one cannot rely upon it to carry any load beyond P_{CR}. In fact, the column will deform laterally so much that the material will fail and the system will collapse. Unlike columns, simply supported plates experience significant stiffening on the secondary path.

4.1.2 Asymmetric Bifurcation

Consider the frame illustrated in Figure 4.2. An eigenvalue analysis using one finite element per bar [24, Sections 5.9 and 7.8] reveals the bifurcation load

$$P_{CR} = 8.932(10^{-6})AE \tag{4.5}$$

but gives no indication about the nature of the critical state: whether it is stable or not, whether the post-critical path is symmetric or not, and so on. We shall see later on that the frame has an asymmetric, and thus unstable, post-critical path, as represented in Figure 4.2. That is, the post-critical path has a slope

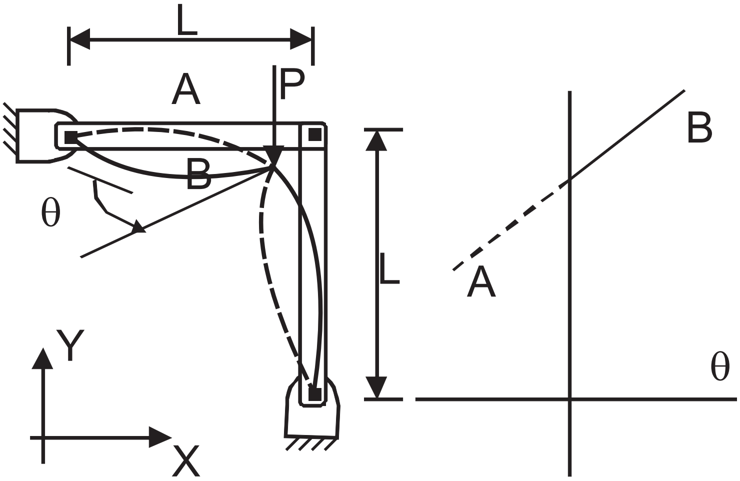


Fig. 4.2: Two bar frame.

$$P^{(1)} = 18.73(10^{-9})AE \text{ (1/rad)} \tag{4.6}$$

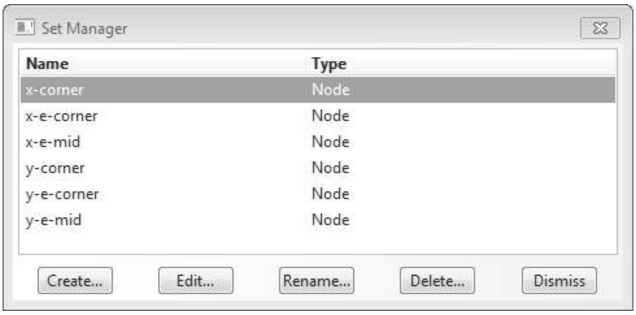
in the force-rotation diagram in Figure 4.2, where θ is the rotation of the joint at the load point.

In general, the problem with eigenvalue analysis is that it provides no indication as to the nature of the post-critical path. If the post-critical path is stiffening and symmetric as in Figure 4.1, the real structure may have a load capacity close to the bifurcation load. But if the post-critical path is unstable and/or asymmetric, as in Figure 4.2 or if there is mode interaction [20, 25–29], the real structure may have a load capacity much smaller than the bifurcation load. In order to use the information provided by eigenvalue analysis, it is necessary to understand and quantify the post-buckling behavior.

4.1.3 Post-critical Path

One way to investigate the post-buckling behavior is to perform a continuation analysis of the imperfect structure, as presented in Section 4.2. This is perfectly possible, but complicated and time consuming, as it will be seen later in this chapter. A more expedient solution can be obtained using software capable of predicting the nature of the post-critical path, including symmetry, curvature, and mode interaction. If the secondary path is stable and symmetric, the bifurcation load can be used as a good estimate of the load capacity of the structure. The curvature of the post-critical path gives a good indication of the post-buckling stiffening and it can be used to a certain extent to predict post-buckling deformations.

The bifurcation load, slope, and curvature of the post-critical path emerging from the bifurcation (4.1) can be computed with BMI3 [26–28] available in [5]. The post-buckling behavior is represented by the following formula



Name	Type
x-corner	Node
x-e-corner	Node
x-e-mid	Node
y-corner	Node
y-e-corner	Node
y-e-mid	Node

Fig. 4.3: Node sets defined after meshing to be used for applying concentrated loads at nodes.

$$\Lambda = \Lambda^{(cr)} + \Lambda^{(1)} s + \frac{1}{2}\,\Lambda^{(2)} s^2 + \ldots \tag{4.7}$$

where s is the perturbation parameter, which is chosen as one component of the displacement of one node, $\Lambda^{(cr)}$ is the bifurcation multiplier, $\Lambda^{(1)}$ is the slope, and $\Lambda^{(2)}$ is the curvature of the post-critical path ([25, (43)], see also [20, 26–29]). When the slope is zero, the post-critical path is symmetric. Therefore, buckling is indifferent, and the real structure will buckle to either side. There is no way to predict which way it is going to buckle, unless of course one knows the shape of the imperfections on the real plate, which is seldom the case. A positive curvature denotes stiffening during post-buckling, and a negative one indicates that the stiffness decreases.

Example 4.2 *Consider the simple supported plate of Example 4.1, p. 179. Compute the bifurcation multiplier $\Lambda^{(cr)}$, the critical load N_{cr}, the slope $\Lambda^{(1)}$, and the curvature $\Lambda^{(2)}$ of the post-critical path. Estimate the load when the maximum lateral deflection is equal to the thickness of the plate. As perturbation parameter, use the largest displacement component of the buckling mode with lowest buckling load.*

Solution to Example 4.2 *The program BMI3 [5] is used in this case to compute the bifurcation multiplier $\Lambda^{(cr)}$, the slope $\Lambda^{(1)}$, and the curvature $\Lambda^{(2)}$ of the post-critical path. Refer to Appendix C for a description of the software interface and operation procedure. Since BMI3 requires the A-B-D-H matrices, we can use the* `.cae` *file generated during the solution of Example 4.1(b) (available as* `Ex_4.1.b.cae` *in [5]), but a few changes are needed to satisfy the current limitations of BMI3, as follows.*

i. Open the `Ex_4.1.b.cae` *and save it as* `Ex_4.2.cae`

```
Menu:   File, Set Work Directory [C:\SIMULIA\user\Ex_4.2], OK
Menu:   File, Open [C:\SIMULIA\user\Ex_4.1\Ex_4.1.b.cae], OK
Menu:   File, Save As [C:\SIMULIA\user\Ex_4.2\Ex_4.2.cae], OK
```

ii. Delete the mesh. Then reseed and remesh with a 2×4 *mesh of S8R5 elements.*

```
Module: Mesh
Object: Assembly
Menu:   Mesh, Delete Instance Mesh, Yes
Menu:   Seed, Edges
```

```
        # pick: horizontal lines, Done
        Method: By Number, Number of Elements [4], OK
        # pick: vertical lines, Done
        Method: By Number, Number of Elements [2], OK
    Menu:   Mesh, Instance, Yes
```

iii. *BMI3 only accepts concentrated loads. Therefore, the distributed edge loads must be replaced by a set of concentrated loads. Since S8R5 are quadratic elements, a uniform (constant) distributed load on the edge is equivalent to three loads concentrated at the nodes. For a unit edge load ($N_n = 1.0\ N/mm$), the element corner nodes get 1/6 each and the mid-node gets 4/6. At a node where 2 elements contribute, the contributions from both elements add up to 1/3. To do this change, first create node sets to later apply the concentrated loads. Six sets are needed as illustrated in the* `Set Manager` *window shown Figure in 4.3. A set is created as follows:*

```
    Module: Load
    Menu:   Tools, Set, Manager
        Create, Name [x-corner], Type: Node, Cont
        # pick: the two corners of the plate on the edge at x=500, Done
```

iv. *Using the same procedure, create the additional sets. The* `x-e-corner` *contains the nodes where pairs of elements join (at the edge at $x = 500\ mm$). The* `x-e-mid` *contains the mid–side nodes of elements on the edge at $x = 500\ mm$. The remaining sets, for the edge at $y = 250\ mm$, are created similarly.*

v. *Delete all the edge loads and replace them by concentrated loads, as follows:*

```
    Module: Load
    Menu:   Load, Manager
        # delete the existing load
        Create, Step: Step-1, Mechanical, Concentrated force, Cont
        Sets, x-corner, Cont, CF1 [-41.67], OK
        Create, Step: Step-1, Mechanical, Concentrated force, Cont
        Sets, x-e-corner, Cont, CF1 [-83.33], OK
        Create, Step: Step-1, Mechanical, Concentrated force, Cont
        Sets, x-e-mid, Cont, CF1 [-166.67], OK
        # Analogously add compressive loads on the edge y=250
        # close the Load Manager
```

vi. *Run the model to verify it produces the right results. Notice the eigenvalues are slightly different than those obtained in Example 4.1; this is due to the coarseness of the mesh used.*

```
    Module: Job
    Menu:   Job, Manager
        Create, Name [Job-1], Cont, OK, Submit, # when Completed, Results
        # visualize the mode shapes as it was indicated in Example 4.1
```

vii. *Copy the filtering program* `inp2bmi3.exe` *(available in [5]) to the* `Work Directory`*. Execute this file by double-clicking on it. It will create the files* `ABQ.inp`*,* `BMI3.inp`*, and* `BMI3.dat`*, based on the file* `Job-1.inp`*. The file* `ABQ.inp` *is a filtered copy that* Abaqus *can execute.*

viii. *Create a new Job to run* `ABQ.inp` *to verify everything went well with the filtering. Note that unlike previous examples where a Job was created to run a model, in this case a Job is created to run an input file. This is done as follows:*

```
Module: Job
Menu:   Job, Manager
    Create, Name [ABQ], Source: Input file # not Model
    Select, ABQ.inp, OK, Cont, OK
    Submit, # when Completed, Results
```

ix. *Copy the program* `bmi3.exe` *(available in [5]) to the* `Work Directory` *and execute it by double-clicking on it. Take note of the following results, either from the display output of from the file* `BMI3.out`*:*

By default, BMI3 chooses as perturbation parameter the largest displacement component of the buckling mode with lowest buckling load, but that can be changed by editing the file `BMI3.dat`. *For this example, the default perturbation corresponds to the first buckling mode, the node in the middle of the plate, and the deflection direction U3 perpendicular to the surface of the plate. The results, specially $\Lambda^{(2)}$, are sensitive to mesh refinement.*

Since BMI3 solves the problem using reversed loads (see Appendix C), then (4.7) becomes

$$\begin{aligned} -N &= \Lambda^{(cr)} + \Lambda^{(1)}s + \tfrac{1}{2}\,\Lambda^{(2)}s^2 \\ N &= -\Lambda^{(cr)} - \Lambda^{(1)}s - \tfrac{1}{2}\,\Lambda^{(2)}s^2 \end{aligned}$$

and, in this case, the perturbation direction is $s = -\delta$, so

$$\begin{aligned} N &= -\Lambda^{(cr)} - \Lambda^{(1)}(-\delta) - \tfrac{1}{2}\,\Lambda^{(2)}(-\delta)^2 \\ N &= -\Lambda^{(cr)} + \Lambda^{(1)}\delta - \tfrac{1}{2}\,\Lambda^{(2)}\delta^2 \end{aligned}$$

Therefore, using the results from a more refined mesh, the secondary path is

$$N = -(-209.0) + (0)\ \delta - (-0.1154)\ \delta^2 = 209.0 + 0.1154\ \delta^2$$

Since the slope $\Lambda(1)$ is zero, the post-critical path is symmetric. The post-buckling load when the lateral deflection (w) is equal to the thickness ($s = Th = 10.2$ mm) is equal to 221 N/mm, as shown in Figure 4.4.

4.2 Continuation Methods

The strain to failure of polymer matrix composites (PMC) is high. Compare 1.29% for AS4/3501 and 2.9% for S-glass/epoxy with only 0.2% for steel and 0.4% for aluminum. That means that buckling deformations can go into a post-buckling regime while the material remains elastic. However, great care must be taken that no matrix dominated degradation mode takes place, in which case the material will not remain elastic (see Chapters 8 and 9). Eigenvalue buckling analysis is relatively simple as long as the material remains elastic because classical theory of elastic stability can be used, as was done in Section 4.1. Material nonlinearity is one reason that motivates an incremental analysis. Another reason is to evaluate the magnitude of the buckling load for an *imperfection sensitive* structure.

In an incremental analysis, also called continuation analysis, the load is increased gradually step by step. At each step, the deformation, and possibly the changing

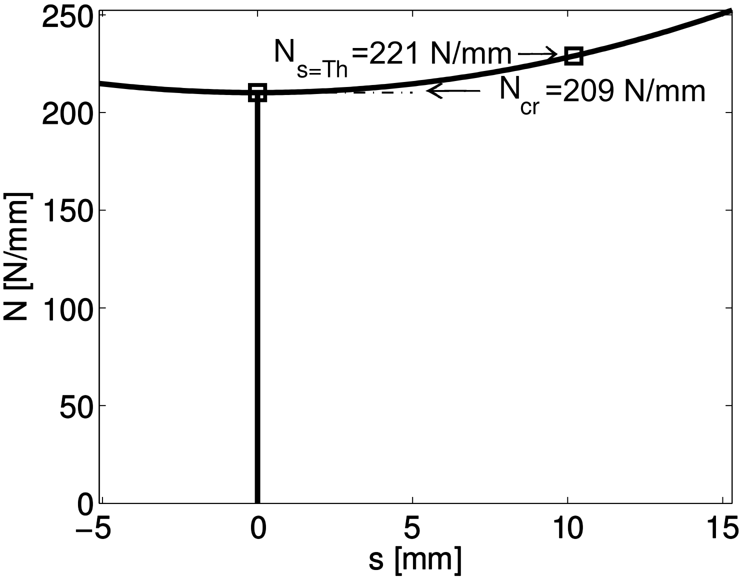


Fig. 4.4: Equilibrium paths for a perfect plate.

material properties, is evaluated. Incremental analysis must include some type of imperfection, in the geometry, material, or alignment of loads. Lacking any imperfection, incremental analysis will track the linear solution, revealing no bifurcations or limit points.

Continuation methods are a form of geometrically nonlinear analysis. The system must have a nontrivial fundamental path, such as a flat plate with asymmetric laminate stacking sequence (LSS) under edge loads.

If the system has a trivial fundamental path, such as a flat plate with symmetric LSS under edge load, the nontrivial fundamental path can be forced by introducing an imperfection. Several types of imperfections are possible, including material imperfections (e.g., unsymmetrical LSS), geometric imperfections, or load eccentricity are used.

Since the real geometric imperfections are seldom known, the preferred artificial geometric imperfection is in the form of the bifurcation mode having the lowest bifurcation load. This is true in most cases; however, in some cases, a second mode that is associated to imperfections that are more damaging to the structure should be used [30]. Also, if the structure has an asymmetric post-buckling path, as the two-bar example in Figure 4.2, care must be taken not to force the structure along the stiffening path.

FEA codes allow the user to modify a mesh by superposing an imperfection in the shape of any mode from a previous bifurcation analysis onto the perfect geometry (see Example 4.3).

Example 4.3 *Apply a geometric imperfection $w_p(x,y) = \delta_0\ \phi(x,y)$ to Example 4.1 (p. 179) and plot the load-multiplier vs. maximum lateral deflection for an imperfection magnitude $\delta_0 = Th/10$ and $\delta_0 = Th/100$, where Th is the total laminate thickness, and $\phi(x,y)$ is the* *buckling mode corresponding to the lowest bifurcation load found in Example 4.1, p. 179.*

Solution to Example 4.3 *The buckling modes were found in Example 4.1, p. 179. Here we will be using model created as* `Ex_4.1.b.cae`*. First, open* `Ex_4.1.b.cae` *available in [5] and save it as* `Ex_4.3.cae`*. Then, write the* `Job-1.inp` *file from the* `Job Manager` *window. The solution procedure is summarized below:*

i. Setting `Work Directory`

```
Menu:   File, Set Work Directory, [C:\SIMULIA\user\Ex_4.3], OK
Menu:   File, Open, [C:\SIMULIA\user\Ex_4.1\Ex_4.1.b.cae], OK
Menu:   File, Save As, [C:\SIMULIA\user\Ex_4.3\Ex_4.3.cae], OK
Module: Job
Menu:   Job, Manager
    Create, Name [Job-1], Cont, OK, Write Input
    # the input file Job-1.inp has been created in the Work Directory
```

ii. Compute and save the mode shapes (displacements) of the first few modes. For this, open `Job-1.inp` *with Notepad++ (or your preferred text editor). Save the file as* `Job-pert.inp`*. Then, add the following lines to the* `*Buckle` *step before the text* `**BOUNDARY CONDITIONS`

```
**Data lines to specify the reference load,
*NODE FILE, GLOBAL=YES, LAST MODE=4
U
**
```

and save the file. This will save the mode shapes of the first 4 modes to the `.fil` *results file. The* `.fil` *file is binary, so you cannot open it with Notepad; it is to be read by* Abaqus *only.*

iii. In Abaqus *CAE, create a new job and submit it as follows:*

```
Module: Job
Menu:   Job, Manager
    Create, Name: [Job-pert], Source: Input file
    Input file: Select, [Job-pert.inp], OK, Cont, OK
    Submit
```

If the execution is completed without errors, the file `Job-pert.fil` *should have been created in the* `Work Directory`*. Since the mode shapes are normalized, the maximum displacement in the results file is* 1.0*.*

iv. Set up a Risk nonlinear analysis.
Using the following procedure, the displacements from a previous analysis can be used to update the geometry of the finite element model. In this example, the multiplier factor for the first mode is chosen to be $\delta_1 = 0.05$. The contributions of modes 2 and 3 are commented out in the `.inp` *file (as denoted by lines starting with double star **). Therefore, an initial deflection equal to the first mode of buckling with a central deflection δ_1 is forced on the structure. To do this, open* `Job-1.inp` *with Notepad++ and save it as* `Job-risk.inp`*. Then, replace the following 3 lines*

```
*Step, name=Step-1, perturbation
*Buckle
10, , 10, 70
```

by the following:

```
**IMPERFECTION, FILE=Job-pert, STEP=n
** n : step number where the modes were calculated in the previous run
**     if there was only a Initial Step followed by a Buckle Step, n=1
*IMPERFECTION, FILE=Job-pert, STEP=1
** Multiple data lines to specify the contributions of each mode
** m, delta_m
** m: mode number
** delta_m : scale factor for mode m
1,0.05
**2,0.02
**3,0.01
*STEP, NLGEOM
** next 2 lines (2nd one is empty) define increment/stopping criteria
*STATIC, RIKS
```

followed by the same multiple data lines to specify reference loading used in previously `Job-pert.inp`*,*

```
**Data lines to specify the reference load,
*NODE FILE, GLOBAL=YES, LAST MODE=4
U
**
```

Be sure to delete *all the lines related to* `load case=2` *and save the file. Otherwise, you will get an error message. The lines to be eliminated are similar to:*

```
*Boundary, op=NEW, load case=2
_PickedSet5, 3, 3
*Boundary, op=NEW, load case=2
_PickedSet6, 1, 1
*Boundary, op=NEW, load case=2
_PickedSet7, 2, 2
```

v. In Abaqus *CAE, create a new job and submit it as follows:*

```
Module: Job
Menu:   Job, Manager
    Create, name: [Job-risk], Source: Input file
    Input file: Select, [Job-risk.inp], OK, Cont, OK
Submit, #  when Completed, Results
    # this finalizes the Risk analysis
Module: Visualization
Menu:   Plot, Contours, On Deformed Shape
    Toolbar: Field Output: Primary, U, U3
Menu:   Result, Step/Frame, Increment 25, Apply
Menu:   Result, Step/Frame, Increment 30, Apply
```

Using a continuation method with this non-perfect geometry, the continuation equilibrium paths shown in Figure 4.5 are obtained. It can be seen that eventually the continuation solution approaches the secondary path of the perfect structure, shown by dashed lines in Figure 4.5. For smaller imperfections, the continuation solution follows more closely the

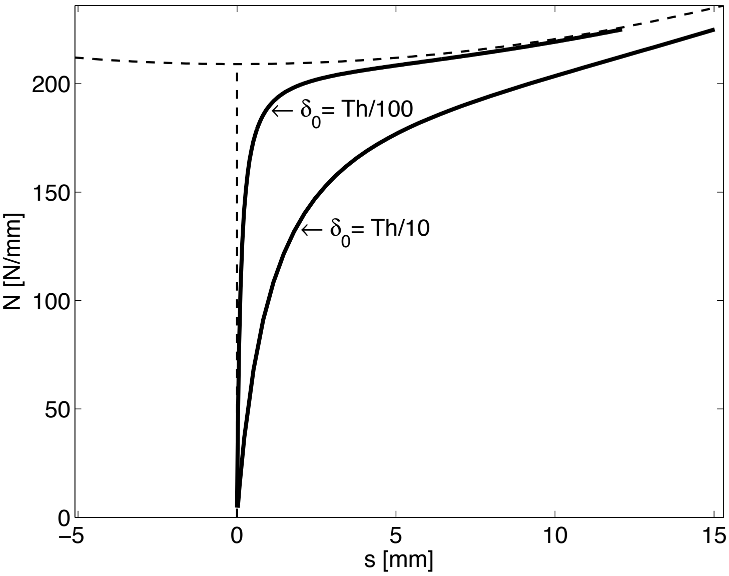


Fig. 4.5: Equilibrium paths for a $[(0/90)_3]_S$ plate, with $\delta_0 = Th/10$ and $\delta_0 = Th/100$.

primary path, then the secondary path. A structure with large imperfections deviates more from the behavior of the perfect structure, as shown by the solution corresponding to an imperfection $\delta_0 = Th/10$.

Example 4.4 *Find the buckling load multiplier and the first mode shape for the column analyzed in Example 3.11, p. 159.*

Solution to Example 4.4 *This example is solved simply modifying the file* `Ex_3.11.cae`*. The dead-load step must be redefined as a buckling step file. The pseudo code below describes the necessary changes.*

i. *Open the* `Ex_3.11.cae` *and save it as* `Ex_4.4.cae`

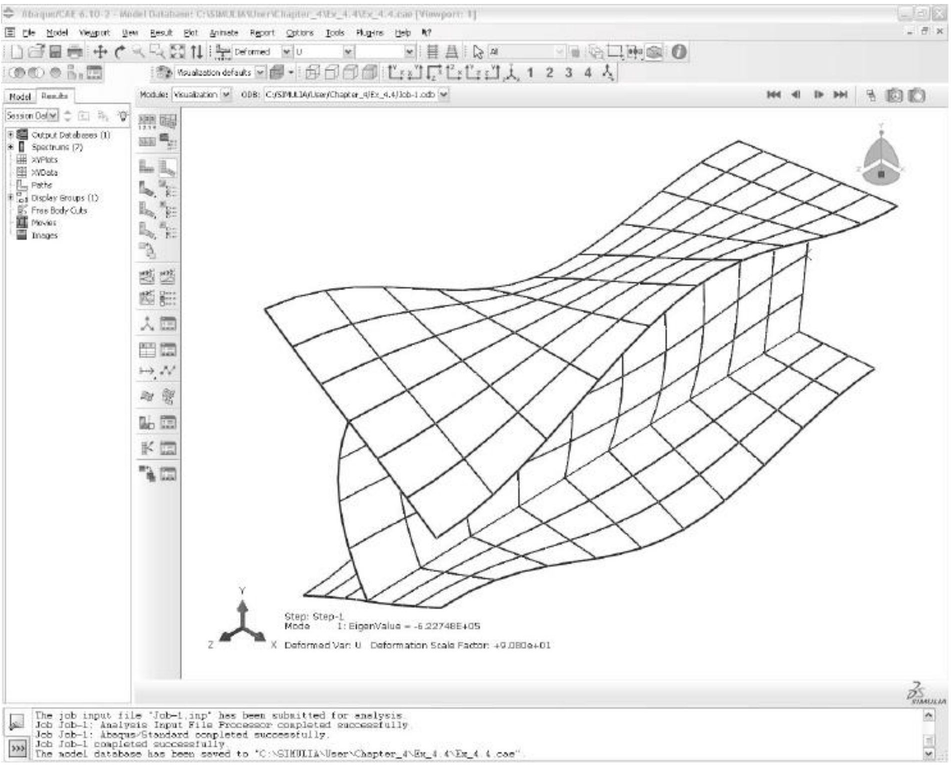
```
Menu:   File, Set Work Directory [C:\SIMULIA\user\Ex_4.4], OK
Menu:   File, Open [C:\SIMULIA\user\Ex_3.11\Ex_3.11.cae], OK
Menu:   File, Save As [C:\SIMULIA\user\Ex_4.4\Ex_4.4.cae], OK
```

ii. *Create a buckling step to replace the dead-weight step used in Example 3.11*

```
Module: Step
Menu:   Step, Manager
    # select: Step-1, Delete, Yes
    Create, Procedure type: Linear perturbation, Buckle, Cont
    Number of eigenvalues requested [1], OK, # close Step Manager
```

iii. *Add a unitary load. The load created to be applied the* `Step-1` *of Example 3.11 was deleted when the analysis step was deleted in the previous instruction.*

```
Module: Load
Menu:   Load, Create
```



```
Step: Step-1
Mode        1: EigenValue = -6.22748E+05
Deformed Var: U   Deformation Scale Factor: +9.080e+01
```

```
The job input file "Job-1.inp" has been submitted for analysis.
Job Job-1: Analysis Input File Processor completed successfully.
Job Job-1: Abaqus/Standard completed successfully.
Job Job-1 completed successfully.
The model database has been saved to "C:\SIMULIA\User\Chapter_4\Ex_4.4\Ex_4.4.cae".
```

Fig. 4.6: Buckling Mode 1 in Example 4.4.

```
    Step: Step-1, Category: Mechanical, Type: Concentrated force, Cont
    # pick: RP-1 in the WS, Done, CF3 [-1.0], OK
```

iv. *Solve the model and visualize the results*

```
Module: Job
Menu:   Job, Manager
    # select: existing job (Job-1), Delete, Yes
    Create, Cont, OK, Submit, # when Completed, Results
Module: Visualization
Menu:   Plot, Deformed Shape
```

In the WS notice that the eigenvalue is indicated as -6.22748E+05. The deformed shape should look similar to the one in Figure 4.6.

Suggested Problems

Problem 4.1 *Compute the bifurcation load P^c of the two-bar frame in Figure 4.2 using one quadratic beam element per bar. Each bar has length $L = 580$ mm, area $A = 41$ mm^2, inertia $I = 8.5$ mm^4, height $H = 10$ mm, and modulus $E = 200$ GPa. The connection between the two bars is rigid.*

Problem 4.2 *Perform a convergence study on the bifurcation load P^c of the two-bar frame in Problem 4.1 by increasing the number of elements per bar N until the bifurcation load converges within 2%. Plot P^c vs. N.*

Problem 4.3 *Recalculate* ***Example*** *4.2 when the LSS changes to $[(0/90)_6]_T$, thus becoming asymmetric. Do not introduce any imperfection but rather analyze the perfect system, which in this case is asymmetric.*

Problem 4.4 *Recalculate* ***Example*** *4.2 with $[(0/90)_6]_T$, and $N_x = 1, N_y = N_{xy} = 0$. Do not introduce any imperfection but rather analyze the perfect system, which in this case is asymmetric.*

Problem 4.5 *Using a FEA code, plot the continuation solution for $\delta_0 = Th/100$ as in Figure 4.5, for a cylindrical shell with distributed axial compression on the edges. The cylinder has a length of $L = 965$ mm and a mid-surface radius of $a = 242$ mm. The LSS is $[(0/90)_6]_S$, with layer thickness $t = 0.127$ mm. The laminas are of E-glass/epoxy with $E_1 = 54\,GPa$, $E_2 = 18\,GPa$, $G_{12} = 9\ GPa$, $\nu_{12} = 0.25$, and $\nu_{23} = 0.38$.*

Problem 4.6 *Compute the maximum stress failure index I_f of Problem 4.5 at $P = \Lambda^{(cr)}$. The strength values are $F_{1t} = 1034$ MPa, $F_{1c} = 1034$ MPa, $F_{2t} = 31$ MPa, $F_{2c} = 138$ MPa, and $F_{6t} = 41$ MPa.*

Problem 4.7 *Plot the imperfection sensitivity of the cylindrical shell of Problem 4.5, for imperfections in the range $(Th/200) < s < Th$.*

Chapter 5

Free Edge Stresses

In-plane loading N_x, N_y, N_{xy} of symmetric laminates induces only in-plane stress σ_x, σ_y, σ_{xy} in the interior of the laminate. Near the free edges, interlaminar stresses σ_z, σ_{yz}, σ_{xz}, are induced due to the imbalance of the in-plane stress components at the free edge.

For illustration, consider a long laminated strip of length $2L$, width $2b << 2L$, and thickness $2H < 2b$ (Figure 5.1). The strip is loaded by an axial force N_x only. For a balanced, symmetric laminate the mid-plane strains and curvatures (see (3.6)) are uniform over the entire cross section and given by

$$\begin{aligned}
\epsilon_x^0 &= \alpha_{11} N_x \\
\epsilon_y^0 &= \alpha_{12} N_x \\
\gamma_{xy}^0 &= 0 \\
k_x &= k_y = k_{xy} = 0
\end{aligned} \tag{5.1}$$

where α_{11}, α_{12} are in-plane laminate compliances, which are obtained by inverting (3.8); see also [1, (6.21)]. From the constitutive equation [1, (6.24)] for lamina k, we get

$$\begin{aligned}
\sigma_x^k &= \left(\overline{Q}_{11}^k \alpha_{11} + \overline{Q}_{12}^k \alpha_{12}\right) N_x \\
\sigma_y^k &= \left(\overline{Q}_{12}^k \alpha_{11} + \overline{Q}_{22}^k \alpha_{12}\right) N_x \\
\sigma_{xy}^k &= \left(\overline{Q}_{16}^k \alpha_{11} + \overline{Q}_{26}^k \alpha_{12}\right) N_x \\
\sigma_z^k &= \sigma_{xz}^k = \sigma_{yz}^k = 0
\end{aligned} \tag{5.2}$$

A piece of laminate taken out of the interior of the laminate will have balanced σ_y and σ_{xy} on opposite faces; the free body diagram (FBD) is in equilibrium without the need for any additional forces. In this case we say the stress components are self-equilibrating. At the free edge in Figure 5.1, σ_y=σ_{xy}=σ_{yz}=0. If σ_y and σ_{xy} are

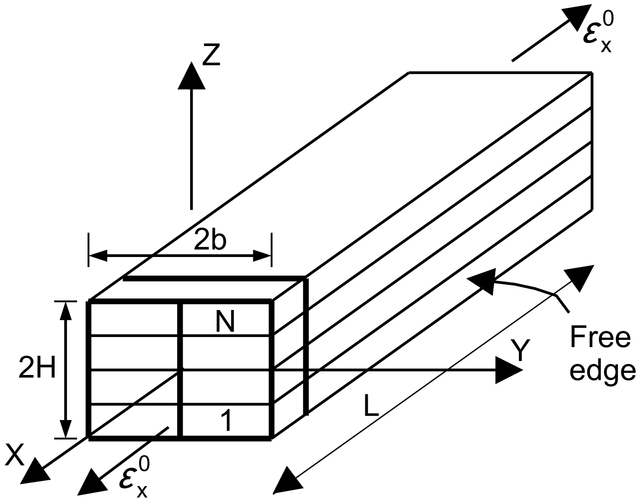


Fig. 5.1: Tensile coupon.[1]

not zero in the interior of the laminate, but are zero at the free edge, then some other stresses must equilibrate them.

5.1 Poisson's Mismatch

A lamina subjected to tensile loading in one direction will contract in the direction perpendicular to the load. If two or more layers with different Poisson's ratios are bonded together, interlaminar stress will be induced to force all layers to deform equally at the interfaces (Figure 5.2). Over the entire laminate thickness, these stresses add up to zero since there is no transverse loading N$_y$ applied. In other words, they are self-equilibrating in such a way that

$$\int_{z_0}^{z_N} \sigma_y dz = 0 \tag{5.3}$$

where z_0 and z_N are the coordinates of the bottom and top surfaces, respectively.

5.1.1 Interlaminar Force

As noted in (5.3), the in-plane stress σ_y calculated with classical lamination theory (CLT, [1, Chapter 6]) is self-equilibrating when added through the whole thickness of the laminate. But on a portion of the laminate (above z_k in Figure 5.3), the stresses σ_y may not be self-equilibrating. Therefore, the contraction or expansion of one or more layers must be equilibrated by interlaminar shear stress σ_{yz}. Since there is no shear loading on the laminate, the integral of σ_{yz} over the entire width of the sample must vanish. Over half the width of the laminate, however, an interlaminar

[1] Reprinted from *Mechanics of Fibrous Composites*, C. T. Herakovich, Fig. 8.1, copyright (1998), with permission from John Wiley & Sons, Inc.

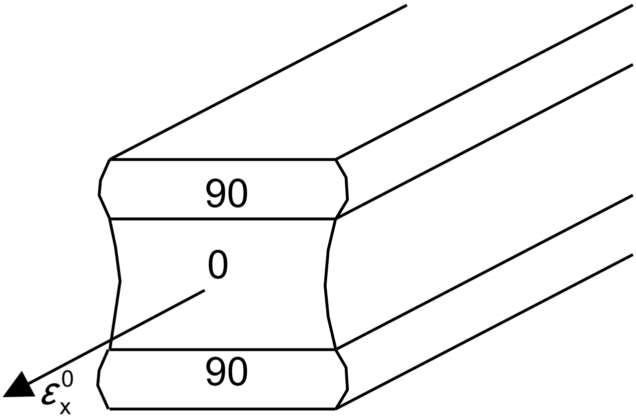


Fig. 5.2: Poisson's effect.[2]

shear force exists if the stress σ_y above or below the surface is not self-equilibrating. The magnitude of these per unit length forces can be estimated by integrating the interlaminar shear stress σ_{yz} over half the width of the laminate ($0 < y < b$). By equilibrium

$$F_{yz}(z_k) = \int_0^b \sigma_{yz_{(z=z_k)}} dy = -\int_{z_k}^{z_N} \sigma_y dz \tag{5.4}$$

The interlaminar shear stress σ_{yz} is not available from classical lamination theory but the transverse stress σ_y is. Therefore, the magnitude of the interlaminar shear force can be computed anywhere through the thickness of a laminate in terms of the known transverse stress distribution σ_y.

The in-plane stress σ_y in a balanced, symmetric laminate under tensile load is constant in each lamina. Therefore, when the interlaminar force is evaluated at an interface (located at z=z$_k$), the integration above reduces to

$$F_{yz}(z_k) = -\sum_{i=k}^{N} \sigma_y^i t_i \tag{5.5}$$

where t_i are the thicknesses of the laminas.

The magnitude of the interlaminar shear force F$_{yz}$ can be used to compare different stacking sequences in an effort to minimize the free-edge interlaminar shear stress σ_{yz}. However, the force does not indicate how large the actual stress is. Therefore, it can be used to compare different LSS but not as a failure criterion.

5.1.2 Interlaminar Moment

The interlaminar shear stress σ_{yz} produces shear strain γ_{yz}, which must vanish at the center line of the sample because of symmetry. Therefore, $\sigma_{yz} = 0$ at the

[2] Reprinted from *Mechanics of Fibrous Composites*, C. T. Herakovich, Fig. 8.14, copyright (1998), with permission from John Wiley & Sons, Inc.

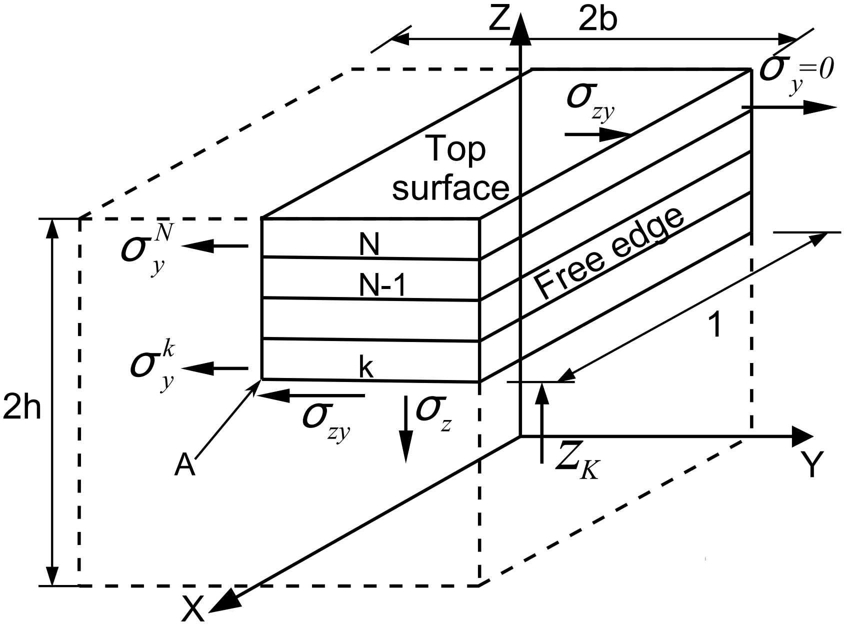


Fig. 5.3: Free body diagram of sublaminate used for computation of Poisson-induced forces F_{yz} and moments M_z.

center line. Also, at the free edge, σ_{yz} must vanish because σ_{zy} vanishes there. But for any position z_k above which σ_y is not self-equilibrating, σ_{yz} must be non-zero somewhere between the edge and the center line. A numerical solution of σ_{yz} is plotted in Figure 5.5 in terms of the distance y/b from the free edge. It reveals that σ_{yz} grows rapidly near the free edge and then tapers out at the interior of the laminate.

A not self-equilibrating distribution of stress yields both a force F$_{yz}$ (5.5) and a moment. To compute the moment M_z, take moments of the stress σ_y with respect to point A in Figure 5.3. A non-vanishing moment produced by σ_y can only be equilibrated by a moment produced by transverse stress σ_z. Therefore, the moment M$_z$ is defined as

$$M_z(z_k) = \int_0^b \sigma_{z_{(z=z_k)}} y dy = \int_{z_k}^{z_N} (z - z_k)\sigma_y dz \tag{5.6}$$

where z_k is the coordinate of the top surface of lamina k, and z_N is the coordinate of the top surface of the laminate (see [1, Figure 6.6] for the definition of the coordinate system through the thickness of the laminate).

The existence of σ_z is corroborated by free-edge delamination during a tensile test, at a much lower load than the failure load of the laminate. The magnitude of the moment can be used to compare different stacking sequences in an effort to minimize the thickness stress σ_z. However, the moment does not indicate how large the actual stress is, and thus it cannot be used as a failure criterion.

The in-plane stress σ_y in a balanced laminate under tensile load is constant in each lamina. Therefore, when the interlaminar moment is evaluated at an interface (located at $z = z_k$), the integration above reduces to

$$M_z(z_k) = \sum_{i=k}^{N} \sigma_y^i (z_i t_i + \frac{t_i^2}{2} - z_k t_i) \tag{5.7}$$

Since σ_z is a byproduct of σ_{yz}, which vanishes at $y = 0$ due to symmetry, then σ_z must vanish at the center line of the specimen ($y = 0$) but it is large near the edge. Since no vertical load is applied, the integral of σ_z must be zero. Therefore, it must be tensile (positive) on some regions and compressive (negative) at others. A numerical solution reveals that σ_z grows rapidly near the free edge, dips to negative values, and then tapers out at the interior of the laminate. A numerical solution of σ_z is plotted in Figure 5.5 in terms of the distance y/b from the free edge. However, $\sigma_z \to \infty$ as $y \to b$. This is a singularity that is not handled well by FEA. Therefore the results, even for $y < b$, will be very dependent on the mesh refinement. Furthermore, since $\sigma_z \to \infty$, the results cannot be used in a failure criterion without further consideration.

Example 5.1 *Compute F_{yz} and M_z at all interfaces of a balanced $[0_2/90_2]_s$ symmetric laminate (Figure 5.1) loaded with $N_x = 175$ KN/m. Use unidirectional lamina carbon/epoxy properties E_1=139 GPa, E_2=14.5 GPa, $G_{12} = G_{13} = 5.86$ GPa, $G_{23} = 5.25$ GPa, $\nu_{12} = \nu_{13} = 0.21$, $\nu_{23} = 0.38$. The lamina thickness is $t_k = 0.127$ mm.*

Solution to Example 5.1 *The in-plane stress distribution σ_y through the thickness can be obtained by the procedure described in [1, Section 6.2], which is implemented in CADEC [12]. The stress values are shown in Table 5.1.*

To calculate F_{yz}, compute the contribution of all layers above a given interface using (5.5). The in-plane stress σ_y in a balanced laminate under in-plane load is constant in each lamina, so (5.5) applies. For other cases, (5.4) can be integrated exactly since σ_y is linear in z, or F_{yz} can be approximated by (5.5) using the average σ_y in each lamina.

Since the laminate is balanced and loaded with in-plane loads only, M_z can be computed using (5.7). Otherwise, use (5.6) or approximate M_z by using the average σ_z in each lamina into (5.7).

The results are shown Table 5.1 and Figure 5.4.

Example 5.2 *Plot the Poisson-induced stresses σ_z and σ_{yz} as a function of y for $0 < y < b$ (Figure 5.3) at the 90/0 interface above the middle surface of a $[0/90]_s$ laminate with properties $E_1 = 139$ GPa, $E_2 = 14.5$ GPa, $\nu_{12} = \nu_{13} = 0.21$, $\nu_{23} = 0.38$, $G_{12} = G_{13} = 5.86$ GPa, $G_{23} = 5.25$ GPa. Take $2b = 20$ mm, length of the sample $2L = 80$ mm, thickness of each lamina $t_k = 1.25$ mm. Load the sample with a uniform strain $\epsilon_x = 0.01$ by applying a uniform displacement at $x = L$. Use orthotropic solid elements on each lamina. Refine the mesh towards the free edge. Use at least two quadratic elements through the thickness of each lamina and an element aspect ratio approximately one near the free edge.*

Solution to Example 5.2 *Note that it is not necessary to model the whole geometry. Symmetry can be used to model only the quadrant with $x > 0, y > 0, z > 0$ (i.e., one-eighth of the plate, Figure 5.3). Since any cross section $y - z$ has the same behavior, only a short segment between $x = 0$ and $x = L\star$ needs to be modeled. Since free edge effects also occur at*

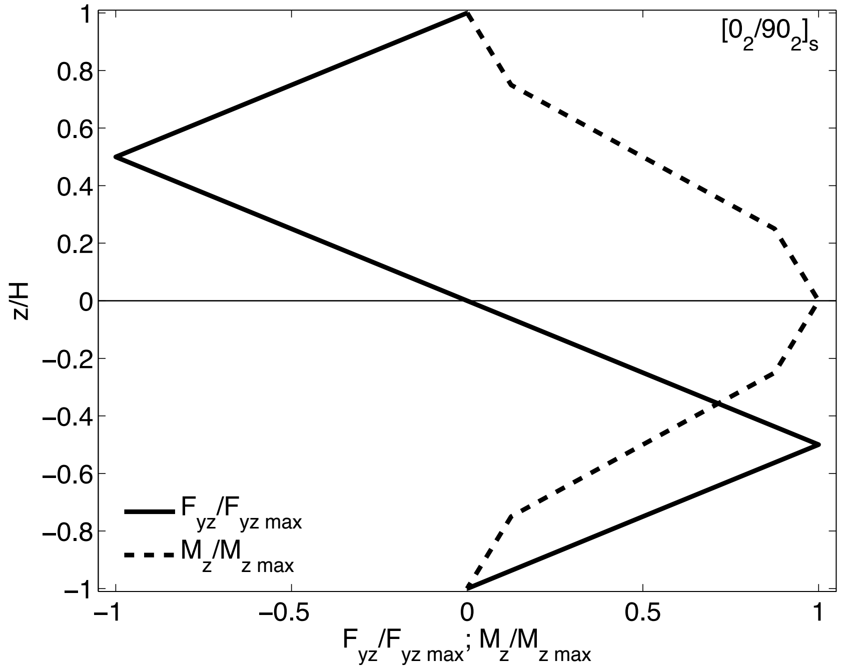


Fig. 5.4: Interlaminar force F_{yz} and moment M_z due to Poisson's effect.

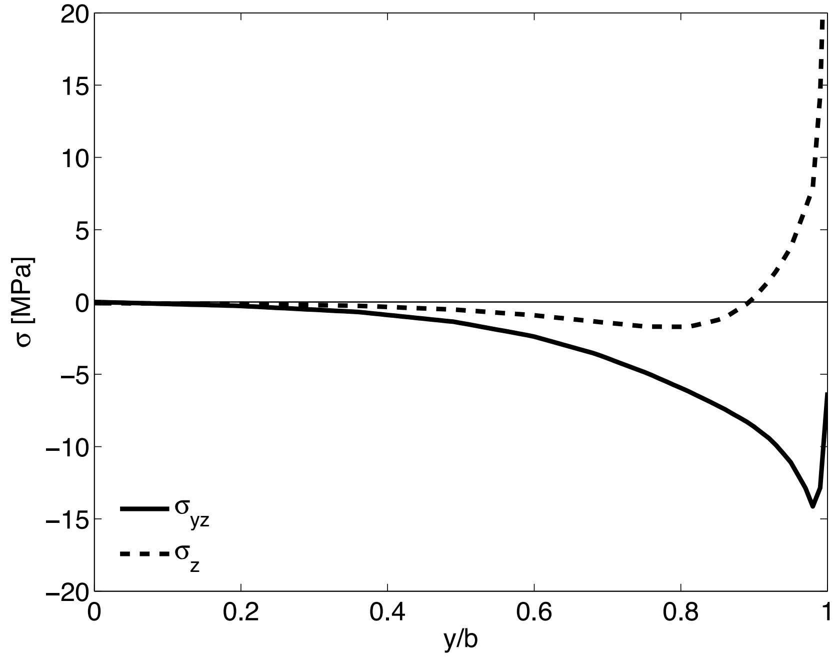


Fig. 5.5: Interlaminar stress σ_{yz} and σ_z at the 90/0 interface of a carbon/epoxy $[0/90]_S$ laminate (FEA).

Table 5.1: Poisson's interlaminar force F_{yz}.

k	Pos	σ_y [MPa]	t_k [mm]	z [mm]	F_{yz} [kN/m]	M_z [N m/m]
8	TOP	$5.55\ 10^{-3}$		0.508	0.000	
8	BOT	$5.55\ 10^{-3}$	0.127	0.381	-0.705	0.045
7	TOP	$5.55\ 10^{-3}$		0.381	-0.705	
7	BOT	$5.55\ 10^{-3}$	0.127	0.254	-1.410	0.179
6	TOP	$-5.55\ 10^{-3}$		0.254	-1.410	
6	BOT	$-5.55\ 10^{-3}$	0.127	0.127	-0.705	0.313
5	TOP	$-5.55\ 10^{-3}$		0.127	-0.705	
5	BOT	$-5.55\ 10^{-3}$	0.127	0.000	0.000	0.358
4	TOP	$-5.55\ 10^{-3}$		0.000	0.000	
4	BOT	$-5.55\ 10^{-3}$	0.127	-0.127	0.705	0.313
3	TOP	$-5.55\ 10^{-3}$		-0.127	0.705	
3	BOT	$-5.55\ 10^{-3}$	0.127	-0.254	1.410	0.179
2	TOP	$5.55\ 10^{-3}$		-0.254	1.410	
2	BOT	$5.55\ 10^{-3}$	0.127	-0.381	0.705	0.045
1	TOP	$5.55\ 10^{-3}$		-0.381	0.705	
1	BOT	$5.55\ 10^{-3}$	0.127	-0.508	0.000	0.000

$x = 0$ and $x = L\star$, take $L\star = 8h = 32t_k$, $t_k = 2.5\ mm$, and plot the results at $x = L\star/2$ to avoid free edge effects at the two loaded ends of the model. The solution is shown in Figure 5.5. See the pseudo code below and Figure 5.5.

i. *Setting* `Work Directory`

```
Menu:   File, Set Work Directory, [C:\SIMULIA\user\Ex_5.2]
Menu:   File, Save As, [C:\SIMULIA\user\Ex_5.2\Ex_5.2.cae]
```

ii. *Creating the part*

```
Module: Part
    # Look at Fig. 5.3, layer thickness is tk=1.25 mm along Z-global
    # b=10 mm, length along x is L=40 mm (shown as 1 in Fig. 5.3)
    # Draw layers 3 and 4 above the middle surface
Menu:   Part, Create
    3D, Deformable, Solid, Extrusion, Approx size [100], Cont
Menu:   Add, Line, Rectangle, [0,0], [40,10], X, Done, Depth [2.5], OK
    # We need to cut the volume in 2 layers
Menu:   Tools, Datum, Plane, Offset from ppal plane, XY Plane, [1.25]
    X, # close the Create Datum pop-up window
Menu:   Tools, Partition, Cell, Use datum plane
    # pick the plane parallel to the XY plane, Create partition, Done
    # close the Create Partition pop-up window
```

iii. *Creating assembly*

```
Module: Assembly
Menu:   Instance, Create, Independent, OK
```

iv. *Defining materials, sections, and assigning sections to parts*

```
Module: Property
Menu:   Material, Create
    Mechanical, Elasticity, Elastic, Type: Engineering Constants
    [1.39E5    1.45E4    1.45E4    0.21    0.21
     0.38      5.86E3    5.86E3    5.2536E3], OK
Menu:   Section, Create, Solid, Homogeneous, Cont, OK
Menu:   Assign, Section, # pick: layer 3, Done, Section-1, OK
    # pick: layer 4, Done, Section-1, OK, Done
    # assign material orientation for 90-deg layer (near mid surface)
Menu:   Assign, Material Orientation, # pick: layer 3, Done
    Use Default Orientation, Definition: Coordinate system
    Additional Rotation Direction: Axis 3
    Additional Rotation: Angle [90], OK
    # assign material orientation for 0-degree layer (top surface)
Menu:   Assign, Material Orientation, #pick layer 4, Done,
    Use Default Orientation, Definition: Coordinate system
    Additional Rotation Direction: Axis 3
    Additional Rotation: None, OK
    # to see the orientations do this:
    # on the left menu tree,
    # Models, Model-1, Parts, Part-1, Orientations
    # you should have only 2 SYSTEM: <Global> entries, one per layer
```

v. *Defining analysis steps*

```
Module: Step
Menu:   Step, Create
    Procedure type: General, Static/General, Cont, OK
```

vi. *Adding loads and BC*

```
Module: Load
Menu:   BC, Manager
    Create, Name [DISP], Step: Step-1, Disp/Rota, Cont
    # pick: surface x=40, Done, # checkmark: U1 [0.4], OK
    # you may need to rotate the model to pick the surfaces
    Create, Name [XSYMM], Step: Initial, Symm/Anti/Enca, Cont
    # pick: surface x=0, Done, XSYMM, OK
    Create, Name [YSYMM], Step: Initial, Symm/Anti/Enca, Cont
    # pick: surface y=0, Done, YSYMM, OK
    Create, Name [ZSYMM], Step: Initial, Symm/Anti/Enca, Cont
    # pick surface z=0, Done, ZSYMM, OK
    # close BC Manager pop-up window
```

vii. *Meshing the model*

```
Module: Mesh
    # Bias towards the edge y=10 is needed
Menu:   Seed, Edges
    # pick: 6 edges parallel to Y-global, Done
    Method: By number, Bias: Single
    Number of elements [10], Bias ratio [20], Apply, OK
    Flip bias, # if required adjust bias direction, point toward y=10
    # pick 6 edges parallel to X-global, Done
    Method: By number, Bias: Double
    Flip bias, # pick: lines, # direct the bias toward the center
    Number of elements: [20], Bias ratio: [10], Apply, OK
    # pick 8 edges parallel to Z-global, Done
    Method: By number, Bias: None, Number of elements [6], OK
Menu:   Mesh, Element Type, # select all regions, Done
    Geometric Order: Quadratic # element C3D20R is assigned, OK
Menu:   Mesh, Instance, Yes
```

viii. *Solving*

```
Module: Job
Menu:   Job, Manager
    Create, Cont, OK, Submit, # when Completed, Results
```

ix. *Visualize stress and strain in **lamina** coordinate system. By default,* Abaqus *displays stress and strain in the lamina coordinate system.*

```
Module: Visualization
Menu:   Plot, Contours, On Deformed Shape
    Toolbox: Field Output, Primary, U, U1  # global coord system
    Toolbox: Field Output, Primary, E, E11 # lamina coord system
    Toolbox: Field Output, Primary, S, S33 # lamina coord system
```

x. *Visualize stress and strain in **global** coordinate system. This method can be used to display stress and strain along any user-defined coordinate system. In this example, we use the fact that one of the layers is oriented in the same direction as the global coordinate system, so the required coordinate system is already available.*

```
Menu:   Result, Options
    Tab: Transformation, Transformation Type: User-specified
    # pick: ASSEMBLY_PART-1-1_ORI-2, Apply, # displays CS, OK
    Toolbox: Field Output, Primary, E, E11 # global coord system
    Toolbox: Field Output, Primary, S, S33 # global coord system
Menu:   Results, Options
    Tab: Transformation, Transformation Type: Default, OK
```

xi. *Create a path for plotting S33 slightly below the 0/90 interface.*

```
Menu:   Tools, Path, Create
    Name [Path-1], Type: Point List, Cont
    [20,0.0,1.24
     20,1.0,1.24
     20,2.0,1.24
     20,3.0,1.24
     20,4.0,1.24
     20,5.0,1.24
     20,6.0,1.24
     20,7.0,1.24
     20,8.0,1.24
     20,9.0,1.24
     20,9.5,1.24
```

```
         20,9.8,1.24
         20,9.9,1.24
         20,10.,1.24]
        # visualize the path created on the WS, OK
```

xii. *Create an XY data set to visualize it and/or save it to a file later.*

```
    Menu:   Tools, XY Data, Create
        Source: Path, Cont, Path: Path-1, Model shape: Undeformed
        X Values: True distance, Y Values: Field output: S, S33, OK
        Save As [path1-S33], OK, Plot, # visualize on screen
        # close XY Data pop-up window
```

xiii. *Save the results on the path to a report file in the Work Directory.*

```
    Menu:   Report, XY
        # pick: path1-S33, Tab: Setup, File name [S33path1.rpt], OK
```

xiv. *Repeat the visualization and report for a path slightly above the interface (defined below).*

```
        [20,0.0,1.26
         20,1.0,1.26
         20,2.0,1.26
         20,3.0,1.26
         20,4.0,1.26
         20,5.0,1.26
         20,6.0,1.26
         20,7.0,1.26
         20,8.0,1.26
         20,9.0,1.26
         20,9.5,1.26
         20,9.8,1.26
         20,9.9,1.26
         20,10.,1.26]
```

Then, save the results in the file [S33path2.rpt]

The stress values along the paths defined above are plotted in Figure 5.6. Notice the results of stress are virtually identical above and below the interface.

5.2 Coefficient of Mutual Influence

In classical lamination theory, it is assumed that the portion of the laminate being analyzed is far from the edges of the laminate. Stress resultants N and M are then applied to a portion of the laminate and these induce in-plane stress σ_x, σ_y, σ_{xy} on each lamina. In the interior of the laminate, interlaminar stress σ_{xz}, σ_{yz} are induced only if shear forces are applied.

For uniaxial loading N_x, the transverse stresses generated in each lamina as a result of Poisson's effect must cancel out to yield a null laminate force N_y. Also, the in-plane shear stress on off-axis layers must cancel out with those of other layers to

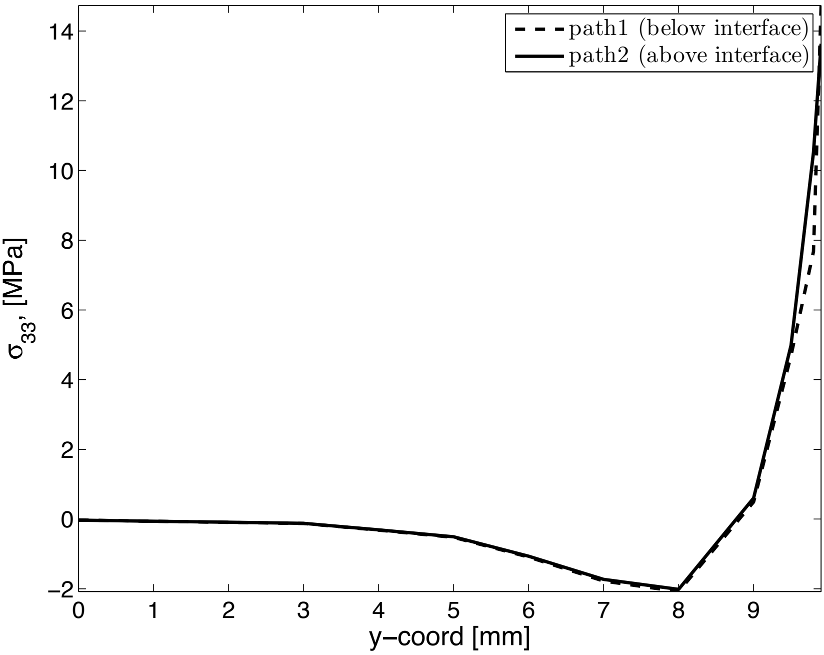


Fig. 5.6: Stress σ_{33} below and above the 90/0 interface.

yield zero shear force N_{xy} for the laminate. The situation is more complex near the edges as the various components of in-plane stress do not cancel each other across the lamina interfaces. For the time being, let us revisit the concept of laminate engineering properties. In material axes, the plane stress compliance equations are

$$\left\{ \begin{array}{c} \epsilon_1 \\ \epsilon_2 \\ \gamma_6 \end{array} \right\} = \left[\begin{array}{ccc} S_{11} & S_{12} & 0 \\ S_{12} & S_{22} & 0 \\ 0 & 0 & S_{66} \end{array} \right] \left\{ \begin{array}{c} \sigma_1 \\ \sigma_2 \\ \sigma_6 \end{array} \right\} \tag{5.8}$$

It is also known that the compliance coefficients can be written in terms of engineering properties as

$$[S] = \left[\begin{array}{ccc} 1/E_1 & -\nu_{12}/E_1 & 0 \\ -\nu_{12}/E_1 & 1/E_2 & 0 \\ 0 & 0 & 1/G_{12} \end{array} \right] \tag{5.9}$$

For an off-axis lamina (oriented arbitrarily with respect to the global axes), we have

$$\left\{ \begin{array}{c} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{array} \right\} = \left[\begin{array}{ccc} \overline{S}_{11} & \overline{S}_{12} & \overline{S}_{16} \\ \overline{S}_{12} & \overline{S}_{22} & \overline{S}_{26} \\ \overline{S}_{16} & \overline{S}_{26} & \overline{S}_{66} \end{array} \right] \left\{ \begin{array}{c} \sigma_x \\ \sigma_y \\ \sigma_{xy} \end{array} \right\} \tag{5.10}$$

Here it can be seen that uniaxial load ($\sigma_y = \sigma_{xy} = 0$) yields shear strain as a result of the shear-extension coupling

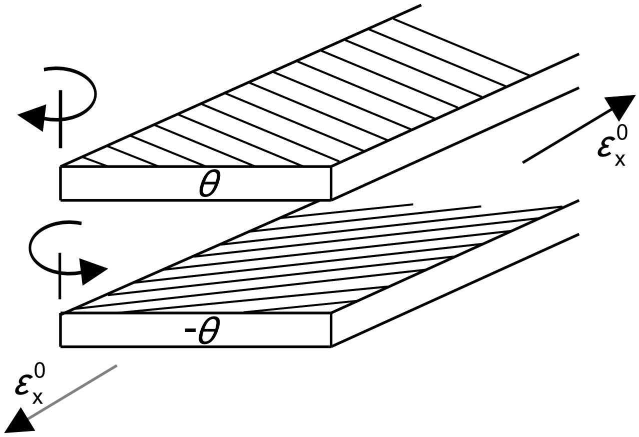


Fig. 5.7: Deformation caused by mutual influence.[3]

$$\gamma_{xy} = \overline{S}_{16}\sigma_x \tag{5.11}$$

where

$$\begin{aligned}\overline{S}_{16} &= (2S_{11} - 2S_{12} - S_{66})\sin\theta\cos^3\theta \\ &- (2S_{22} - 2S_{12} - S_{66})\sin^3\theta\cos\theta\end{aligned} \tag{5.12}$$

Now, the coefficients of $\left[\overline{S}\right]$ can be defined in term of the engineering properties for the off-axis lamina as

$$\begin{aligned}&\overline{S}_{11} = 1/E_x \ ; \ \overline{S}_{12} = -\nu_{xy}/E_x = -\nu_{yx}/E_y \\ &\overline{S}_{22} = 1/E_y \ ; \ \overline{S}_{66} = 1/G_{xy}\end{aligned} \tag{5.13}$$

To complete the definition of $\left[\overline{S}\right]$ in (5.10), two new engineering properties describing shear-extension coupling, $\eta_{xy,x}$ and $\eta_{xy,y}$, are defined as

$$\overline{S}_{16} = \frac{\eta_{xy,x}}{E_x} \quad ; \quad \overline{S}_{26} = \frac{\eta_{xy,y}}{E_x} \tag{5.14}$$

The engineering properties $\eta_{xy,x}$ and $\eta_{xy,y}$ are called coefficients of mutual influence and they represent the shear caused by stretching. Their formal definition is obtained by imposing an axial stress and measuring the resulting shear strain

$$\eta_{ij,i} = \frac{\gamma_{ij}}{\epsilon_i} \tag{5.15}$$

Alternatively, two other coefficients of mutual influence could be defined to represent the stretching caused by shear

[3] Reprinted from *Mechanics of Fibrous Composites*, C. T. Herakovich, Fig. 8.14, copyright (1998), with permission from John Wiley & Sons, Inc.

$$\overline{S}_{16} = \frac{\eta_{x,xy}}{G_{xy}} \; ; \overline{S}_{26} = \frac{\eta_{y,xy}}{G_{xy}} \tag{5.16}$$

These are defined by imposing a shear stress and measuring the axial strain

$$\eta_{i,ij} = \frac{\epsilon_i}{\gamma_{ij}} \tag{5.17}$$

5.2.1 Interlaminar Stress due to Mutual Influence

Off-axis laminas induce in-plane shear stress when subject to axial loading because the natural shear deformations that would occur on an isolated lamina (Figure 5.7) are constrained by the other laminas. Through the whole thickness of the laminate, these stresses cancel out, but over unbalanced sublaminates (e.g., the top lamina in Figure 5.7), they amount to a net shear.

That shear can only be balanced by interlaminar stress σ_{zx} at the bottom of the sublaminate (Figure 5.8). Then, summation of forces along x leads to a net force

$$F_{xz}(z_k) = \int_0^b \sigma_{zx_{(z=z_k)}} dy = -\int_{z_k}^{z_N} \sigma_{xy} dz \tag{5.18}$$

Once again, the in-plane shear stress calculated with classical lamination theory (CLT) [1, Chapter 6] can be used to compute the interlaminar force per unit length F_{xz}. For in-plane loading, CLT yields constant shear stress in each lamina. When the interlaminar force is evaluated at an interface (located at $z = z_k$), the integration above reduces to

$$F_{xz}(z_k) = -\sum_{i=k}^{N} \sigma^i_{xy} t_i \tag{5.19}$$

The force F_{xz}, as well as the values of the coefficients of mutual influence, can be used to qualitatively select the LSS with the least interlaminar stress. Actual values of interlaminar stresses can be found by numerical analysis. However, $\sigma_z \to \infty$ as $y \to b$. This is a singularity that is not handled well by FEA. Therefore the results, even for $y < b$, are very dependent on mesh refinement. Furthermore, since $\sigma_z \to \infty$, the results cannot be used in a failure criterion without further consideration. A numerical approximation of σ_{xz} for a $[\pm 45]_S$ laminate is plotted in Figure 5.10 in terms of the distance y' from the free edge.

Example 5.3 *Compute F_{xz} at all interfaces of a $[30_2/-30_2]_s$ balanced symmetric laminate (Figure 5.1) loaded with $N_x = 175$ KN/m. The material properties are given in Example 5.1, p. 199. The lamina thickness is $t_k = 0.127$ mm.*

Solution to Example 5.3 *In-plane shear stress σ_{xy} through the thickness of the laminate can be obtained following the same procedure used to obtain σ_y in Example 5.1, p. 199.*

For a symmetric balanced laminate under in-plane loads, use (5.19). For a general laminate under general load, use (5.18) or approximate F_{xz} by (5.19) taking the average of σ_{xy} in each lamina.

The results are obtained with a spreadsheet and shown in Table 5.2 and Figure 5.9.

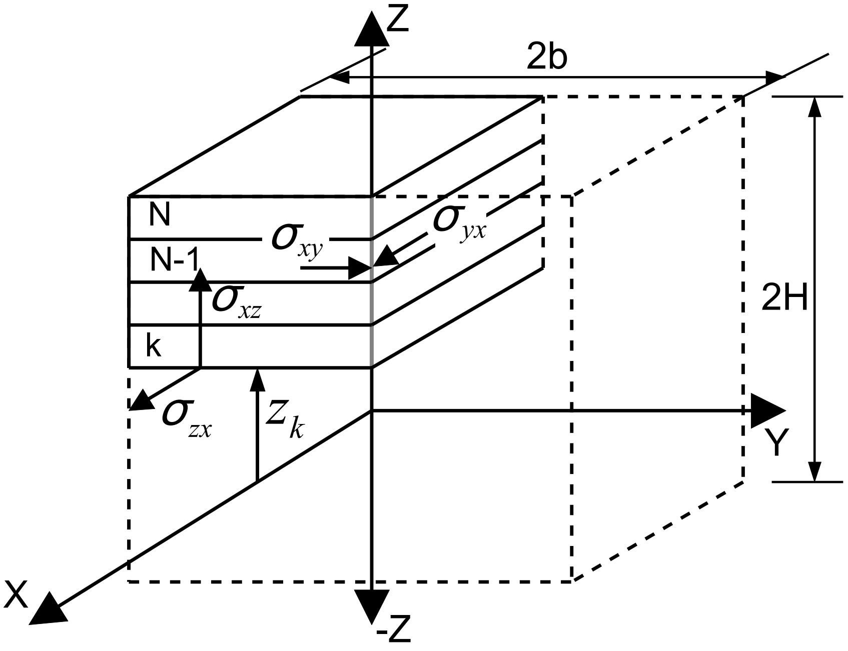


Fig. 5.8: Free body diagram of sublaminate to compute the interlaminar force F_{xz} due to mutual influence.

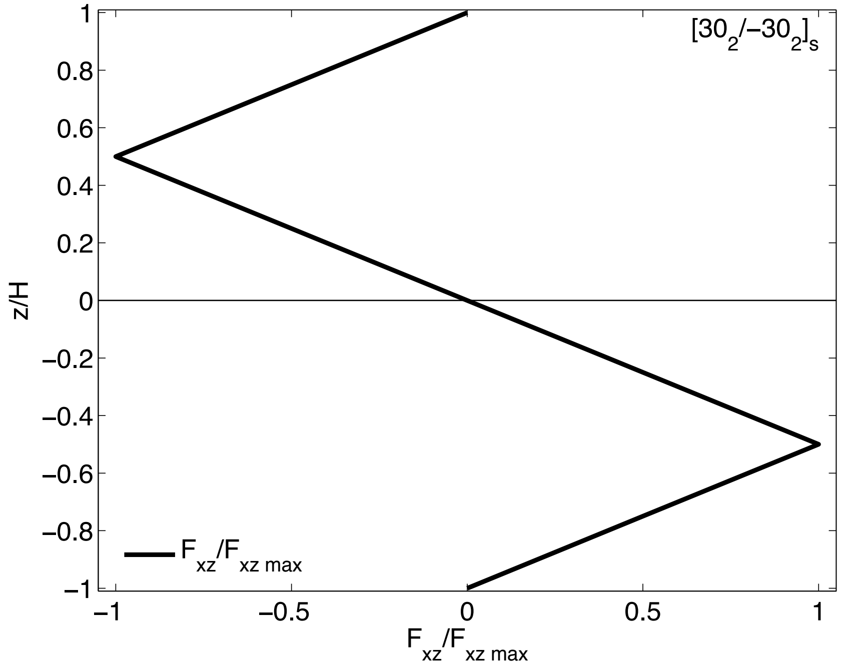


Fig. 5.9: Interlaminar shear force due to mutual influence F_{xz}.

Table 5.2: Interlaminar force F_{xz} due to mutual influence.

k	Pos	σ_{xy}	t_k	z	F_{xz}
		[MPa]	[mm]	[mm]	[kN/m]
8	TOP	$78.6\ 10^{-3}$		0.508	0.000
8	BOT	$78.6\ 10^{-3}$	0.127	0.381	-9.982
7	TOP	$78.6\ 10^{-3}$		0.381	-9.982
7	BOT	$78.6\ 10^{-3}$	0.127	0.254	-19.964
6	TOP	$-78.6\ 10^{-3}$		0.254	-19.964
6	BOT	$-78.6\ 10^{-3}$	0.127	0.127	-9.982
5	TOP	$-78.6\ 10^{-3}$		0.127	-9.982
5	BOT	$-78.6\ 10^{-3}$	0.127	0.000	0.000
4	TOP	$-78.6\ 10^{-3}$		0.000	0.000
4	BOT	$-78.6\ 10^{-3}$	0.127	-0.127	9.982
3	TOP	$-78.6\ 10^{-3}$		-0.127	9.982
3	BOT	$-78.6\ 10^{-3}$	0.127	-0.254	19.964
2	TOP	$78.6\ 10^{-3}$		-0.254	19.964
2	BOT	$78.6\ 10^{-3}$	0.127	-0.381	9.982
1	TOP	$78.6\ 10^{-3}$		-0.381	9.982
1	BOT	$78.6\ 10^{-3}$	0.127	-0.508	0.000

Example 5.4 *Plot σ_{xz} at the interface above the middle surface of a $[\pm 45]_S$ laminate using the material properties, geometry, and loading of Example 5.2, p. 199.*

Solution to Example 5.4 *Since the LSS is symmetric, it is possible to model half of the laminate ($z > 0$). However, it is not possible to use the symmetry conditions used in Example 5.2 (p. 199) because the material (say, a 30^o lamina) is not symmetric with respect to the $x - z$ and $y - z$ planes (see Figure 5.8). Instead, the plane $y - z$ at $x = 0$ is not a symmetry plane but rather a plane with $\epsilon_x = 0$. Also, the edge effects at the ends of the model at $x = 0$ and in $x = L\star$ are now important, so the results must be plotted at $x = L\star/2$ to avoid free edge effects at the loaded ends. The solution is shown in Figure 5.10.*

i. *Setting the* `Work Directory`

```
Menu: File, Set Work Directory, [C:\SIMULIA\user\Ex_5.4]
Menu: File, Save As, [C:\SIMULIA\user\Ex_5.4\Ex_5.4.cae]
```

ii. *Creating the part*

```
Module: Part
Menu:   Part, Create
    3D, Deformable, Solid, Extrusion, Approx size [100], Cont
Menu:   Add, Line, Rectangle, [0,0], [80,20], X, Done, Depth [2.5], OK
    # We need to cut the volume in 2 layers
Menu:   Tools, Datum, Plane,
    Offset from principal plane, XY Plane, [1.25]
    X, # close the pop-up window
Menu:   Tools, Partition, Cell, Use datum plane
    # pick the plane parallel to the XY plane, Create partition, Done
    # close pop-up window
```

iii. *Creating assembly*

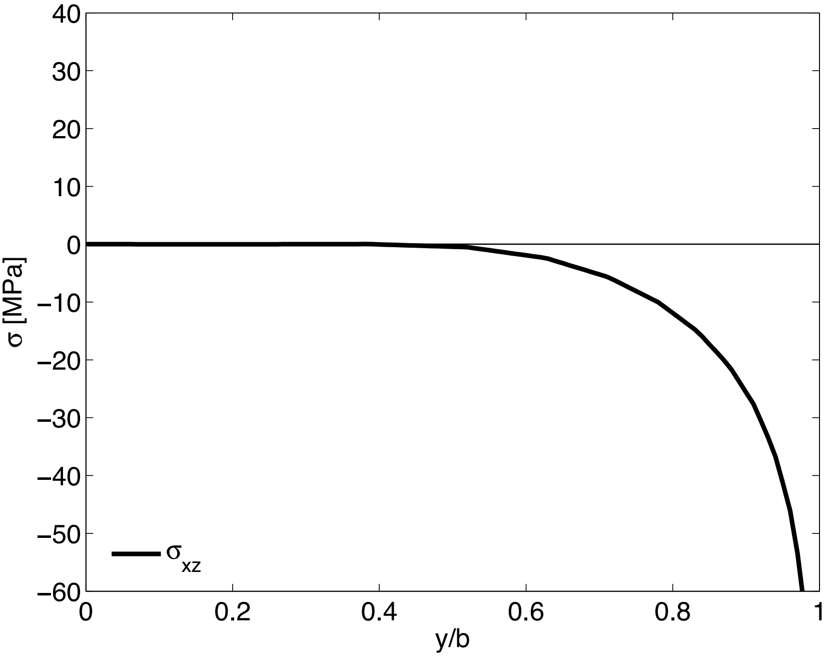


Fig. 5.10: Interlaminar shear stress σ_{xz} at the interface above the middle-surface of a carbon/epoxy $[\pm 45]_S$ laminate (FEA).

```
Module: Assembly
Menu:   Instance, Create, Independent, OK
```

iv. Defining materials, sections, and assigning sections to parts

```
Module: Property
Menu:   Material, Create
    Mechanical, Elasticity, Elastic, Type: Engineering Constants
    [1.39E5    1.45E4    1.45E4    0.21    0.21
     0.38      5.86E3    5.86E3    5.2536E3], OK
Menu:   Section, Create, Solid, Homogeneous, Cont, OK
Menu:   Assign, Section, # pick: layer 3, Done, Section-1, OK
    # pick: layer 4, Done, Section-1, OK, Done
    # assign material orientation -45 layer (near middle surface)
Menu:   Assign, Material Orientation, # pick: layer 3, Done
    Use Default Orientation, Definition: Coordinate system
    Additional Rotation Direction: Axis 3
    Additional Rotation: Angle [-45], OK
    # assign material orientation for +45-degree layer (top surface)
Menu:   Assign, Material Orientation, #pick layer 4, Done,
    Use Default Orientation, Definition: Coordinate system
    Additional Rotation Direction: Axis 3
    Additional Rotation: Angle [45], OK
    # to see the orientations do this:
    # on the left menu tree,
    # Models, Model-1, Parts, Part-1, Orientations
    # you should have only 2 SYSTEM: <Global> entries, one per layer
```

v. Defining analysis steps

```
Module: Step
Menu:   Step, Create
    Procedure type: General, Static/General, Cont, OK
```

vi. Adding loads and BC

```
Module: Load
Menu:   BC, Manager
    Create, Name [DISP], Step: Step-1, Disp/Rota, Cont
    # pick: surface x=80, Done, # checkmark: U1 [0.8], OK
    # you may need to rotate the model to pick the surfaces
    Create, Name [XEND], Step: Initial, Rota/Disp, Cont
    # pick: surface x=0, Done, # checkmark: U1, OK
    Create, Name [ZSYMM], Step: Initial, Symm/Anti/Enca, Cont
    # pick: surface z=0, Done, # checkmark: U3, OK
    # close BC Manager pop-up window
```

vii. Meshing the model

```
Module: Mesh
Menu:   Seed, Edges
    # pick: 6 edges parallel to Y-global, Done
    Method: By number, Bias: Double
    # make sure the bias is directed toward the end of the lines
    Number of elements [20], Bias ratio [10], Apply, OK
    # pick 6 edges parallel to X-global, Done
    Method: By number, Bias: Double
    Flip bias, # all the lines toward the center
    Number of elements: [20], Bias ratio: [10], Apply, OK
    # pick 8 edges parallel to Z-global, Done
    Method: By number, Bias: None, Number of elements [6], Apply, OK
Menu:   Mesh, Element Type, # select all regions, Done
    Geometric Order: Quadratic, # C3D20R is assigned, OK
Menu:   Mesh, Instance, Yes
```

viii. Solving and visualizing the results

```
Module: Job
Menu:   Job, Manager
    Create, Cont, OK, Submit, # when Completed, Results
Module: Visualization
Menu:   Plot, Contours, On Deformed Shape
# create a path for plotting S13 slightly above the -45/+45 interface
Menu:   Tools, Path, Create
    Name [Path-1], Type: Point List, Cont
    [40, 0.0,  1.26
     40, 0.5,  1.26
     40, 1.0,  1.26
     40, 1.5,  1.26
     40, 2.0,  1.26
     40, 2.5,  1.26
     40, 3.0,  1.26
     40, 3.5,  1.26
     40, 4.0,  1.26
     40, 4.5,  1.26
     40, 5.0,  1.26
     40, 6.0,  1.26
     40, 7.0,  1.26
     40, 8.0,  1.26
     40, 9.0,  1.26
     40, 10.0, 1.26
     40, 11.0, 1.26
     40, 12.0, 1.26
     40, 13.0, 1.26
     40, 14.0, 1.26
     40, 15.0, 1.26
     40, 15.5, 1.26
     40, 16.0, 1.26
     40, 16.5, 1.26
     40, 17.0, 1.26
     40, 17.5, 1.26
     40, 18.0, 1.26
     40, 18.5, 1.26
     40, 19.0, 1.26
     40, 19.5, 1.26
     40, 20.0, 1.26]
    # visualize the path created on the WS, OK
Menu:   Tools, XY Data, Create
    Source: Path, Cont, Path: Path-1, Model shape: Undeformed
    X Values: True distance, Y Values: Field output: S, S13
    Save As [S13plot], OK, Plot, # visualize result on screen
    # close XY Data pop-up window
Menu:   Report, XY
    # pick: S13plot, Tab: Setup, File name [S13plot.rpt], OK
    # the file S13plot.rpt must have been saved in the Work Directory
```

The stress values along the path are plotted in Figure 5.11.

Suggested Problems

Problem 5.1 *Write a computer program to use tabulated data of σ_y and σ_{xy} (at the top and bottom of every lamina) to compute F_{yz}, F_{xz}, and M_z, for all locations through the thickness of a laminate with any number of layers. Using the program, plot F_{yz}, F_{xz}, and M_z, through the thickness $-4t < z < 4t$ of a $[\pm 45/0/90]_s$ laminate with lamina thickness $t = 0.125$ mm, loaded with $N_x = 100$ kN/m. Use carbon/epoxy properties E_1=139 GPa, E_2=14.5 GPa, $G_{12} = G_{13} = 5.86$ GPa, $G_{23} = 5.25$ GPa, $\nu_{12} = \nu_{13} = 0.21$, $\nu_{23} = 0.38$. Submit a report including the source code of the program.*

Problem 5.2 *Repeat Problem 5.1 for $M_x = 1$ Nm/m. Submit a report including the source code of the program.*

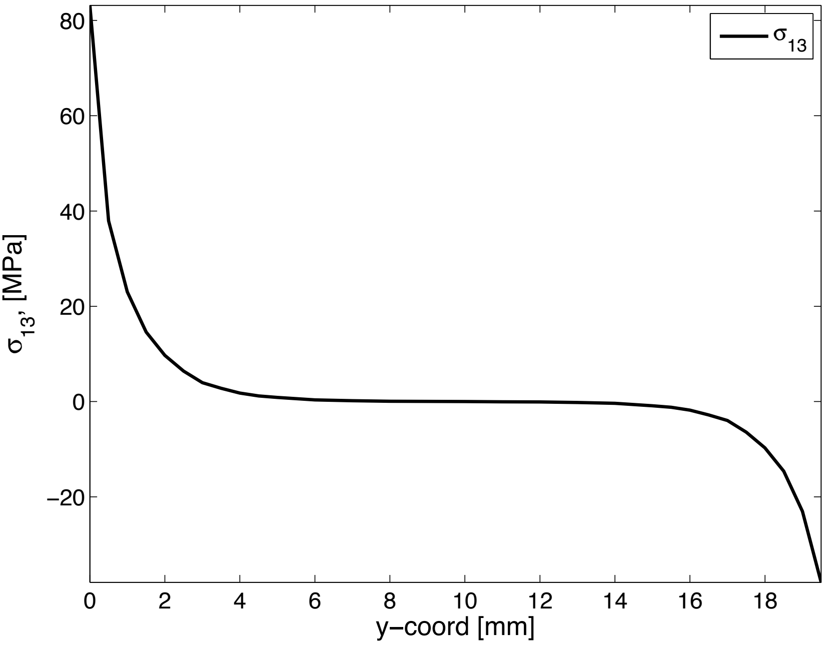


Fig. 5.11: Stress σ_{13} below and above the $-45/+45$ interface.

Problem 5.3 *Plot σ_z/σ_{x0} and σ_{yz}/σ_{x0} vs. y/b $(0 < y/b < 1)$ at $x = L/2$, at the first interface above the mid-surface of a [0/90/90/0] laminate with lamina thickness $t = 0.512$ mm, loaded with $\epsilon_x = 0.01$. Compute the far-field uniform stress σ_{x0} in terms of the applied strain. Use quadratic solid elements and a mesh biased toward the free edge (bias 0.1) to model 1/8 of a tensile specimen (see Example 5.2, p. 199), of width $2b = 25.4$ mm and length $2L = 20$ mm. Use carbon/epoxy properties $E_1 = 139$ GPa, $E_2 = 14.5$ GPa, $G_{12} = G_{13} = 5.86$ GPa, $G_{23} = 5.25$ GPa, $\nu_{12} = \nu_{13} = 0.21$, $\nu_{23} = 0.38$. Attempt to keep the aspect ratio of the elements near the free edge close to one. Submit the input command file to obtain the solution and the plot. In addition, submit the plot.*

Problem 5.4 *For the laminate and loading described in Problem 5.3, plot σ_z/σ_{x0} and σ_{yz}/σ_{x0} versus z/t_k $(0 < z/t_k < 2)$ above the mid-surface, at a distance $0.1t_k$ from the free edge and $x = L/2$. Study the effect of mesh refinement by providing four curves with different number of divisions along the z-direction. Attempt to keep the aspect ratio of the elements near the free edge close to one. Submit the input command file to obtain the solution and the plot. In addition, submit the plot.*

Problem 5.5 *Plot σ_{xz}/σ_{x0} as in Problem 5.3 for all the interfaces above the middle surface of a $[\pm 10_2]_S$ laminate.*

Problem 5.6 *Plot σ_{xz}/σ_{x0} as in Problem 5.4 for a $[\pm 10_2]_S$ laminate.*

Problem 5.7 *Use solid elements and a biased mesh to model 1/8 of a tensile specimen (see Example 5.2, p. 199), of width $2b = 24$ mm and length $2L = 20$ mm. The laminate is $[\pm 45/0/90]_s$ with lamina thickness $t = 0.125$ mm, loaded with $N_x = 175$ KN/m. Use carbon/epoxy properties $E_1 = 139$ GPa, $E_2 = 14.5$ GPa, $G_{12} = G_{13} = 5.86$ GPa, $G_{23} = 5.25$ GPa, $\nu_{12} = \nu_{13} = 0.21$, $\nu_{23} = 0.38$. Plot the three interlaminar stress components,* *from the edge to the center line of the specimen, at the mid-surface of each lamina. Lump all four plots of the same stress into a single plot. Submit the input command file to obtain the solution and the three plots. In addition, submit the three plots.*

Problem 5.8 *Plot E_x/E_2, G_{xy}/G_{12}, $10\nu_{xy}$, $-\eta_{xy,x}$, and $-\eta_{x,xy}$ in the same plot vs θ in the range $-\pi/2 < \theta < \pi/2$ for a unidirectional single lamina oriented at an angle θ. The material is S-glass/epoxy [1, Tables 1.3–1.4].*

Problem 5.9 *Using the plot from Problem 5.8 and considering a $[\theta_1/\theta_2]_S$ laminate, what is the worst combination of values θ_1, θ_2 for two cases: (a) Poisson's mismatch and (b) shear mismatch.*

Problem 5.10 *In a single plot, compare -$\eta_{xy,x}$ of E-glass/epoxy, Kevlar49/epoxy, and T800/3900-2 in the range -$\pi/2 < \theta < \pi/2$ ([1, Tables 1.3–1.4]).*

Problem 5.11 *Obtain contour plots of the three deformations u_x, u_y, u_z (independently) on the top surface of a $[\pm 45]_s$ laminate. Use dimensions, load, and material properties of Problem 5.7. Explain your findings.*

Problem 5.12 *Repeat Problem 5.11 for a $[0/90]_s$ laminate. Explain your findings.*

Problem 5.13 *Use solid elements and a biased mesh to model 1/4 of a tensile specimen (Figure 5.1) of a total width $2b = 12$ mm and length $2L = 24$ mm. Compare in the same plot σ_z vs z/H for $[\pm 15/\pm 45]_s$ and [$\pm$(15/45)] of SCS-6/aluminum with 50% fiber volume. Use micromechanics (6.8) to predict the unidirectional composite properties. The lamina thickness $t_k = 0.25$ mm. The laminate is loaded with $\epsilon_x = 0.01$.*

	Al-2014-T6	*SCS-6*
	[31, App. B]	*[1, Tables 2.1–2.2]*
E [GPa]	*75.0*	*427.0*
G [GPa]	*27.0*	*177.9*

Problem 5.14 *Use a FEA model similar to Problem 5.13 to plot $\sigma_{xz}/\sigma_{zx_{\max}}$ vs θ ($0 < \theta < \pi/2$) for a $[\pm\theta]_s$ SCS-6/ Al laminate with $\epsilon_x = 0.01$.*

Problem 5.15 *Use the FEA model of Problem 5.13 to plot σ_z vs y/b ($0 < y < 0.95b$) at the mid-surface of the $[\pm 15/\pm 45]_s$ laminate. Note $\sigma_z \to \infty$ near $y = b$, so the actual value from FEA at $y = b$ is mesh dependent. Investigate mesh dependency at $y = 0.95b$ by tabulating the result using different mesh densities.*

Problem 5.16 *Use an FEA model similar to Problem 5.13 to plot σ_x, σ_{xy}, and σ_{xz} vs y/b ($0 < y < b$) when a $[\pm\theta]_s$ SCS-6/Al laminate is subjected to 1% axial strain ($\epsilon_x = 0.01$).*

Problem 5.17 *A $[0/90]_s$ laminate with properties $E_1 = 139$ GPa, $E_2 = 14.5$ GPa, $G_{12} = G_{13} = 5.86$ GPa, $G_{23} = 5.25$ GPa, $\nu_{12} = \nu_{13} = 0.21$, $\nu_{23} = 0.38$ is shown in Figure 5.1. The strength properties of the lamina are $F_{1t} = 1550$ MPa, $F_{1c} = 1090$ MPa, $F_{2t} = F_{2c} = 59$ MPa, and $F_6 = 75$ MPa. Take $2b = 20$ mm, length of the sample $2L = 200$ mm, thickness of each lamina $t_k = 1.25$ mm. Load the sample with a uniform strain $\epsilon_x = 0.01$ by applying a uniform displacement. Use symmetry to model only the quadrant with $x > 0$, $y > 0$, $z > 0$. Use orthotropic solid elements on each lamina, with at least two quadratic elements through the thickness of each lamina. Compute the 3D Tsai-Wu failure index I_F using a UMAT subroutine for solid elements. Obtain the contour plot of I_F in each lamina (do not use results averaging). Show all work in a report.*

Chapter 6

Computational Micromechanics

In Chapter 1, the elastic properties of composite materials were assumed to be available in the form of elastic modulus E, shear modulus G, Poisson's ratio ν, and so on. For heterogeneous materials such as composites, a large number of material properties are needed, and experimental determination of these many properties is a tedious and expensive process. Furthermore, the values of these properties change as a function of the volume fraction of reinforcement and so on. An alternative, or at least a complement to experimentation, is to use homogenization techniques to predict the elastic properties of the composite in terms of the elastic properties of the constituents (matrix and reinforcements). Since homogenization models are based on more or less accurate modeling of the microstructure, these models are also called micromechanics models, and the techniques used to obtain approximate values of the composite's properties are called micromechanics methods or techniques [1]. Micromechanics models can be classified into empirical, semiempirical, analytical, and numerical. Accurate semiempirical models are described in [1].

This book deals only with strictly analytical or numerical models that do not require empirical adjusting factors, so that no experimentation is required. Since most of this book deals with 3D analysis, emphasis is placed on micromechanics models that can estimate the whole set of elastic properties using a single model, rather than using a disjoint collection of models based on different assumptions to assemble the set of properties needed. Many analytical techniques of homogenization are based on the equivalent eigenstrain method [32, 33], which considers the problem of a single ellipsoidal inclusion embedded in an infinite elastic medium. The Eshelby solution is used in [34] to develop a method that takes into account, approximately, the interactions among the inclusions. One of the more used homogenization techniques is the self-consistent method [35], which considers a random distribution of inclusions in an infinite medium. The infinite medium is assumed to have properties equal to the unknown properties sought. Therefore, an iterative procedure is used to obtain the overall moduli. Homogenization of composites with periodic microstructure has been accomplished by using various techniques including an extension of the Eshelby inclusion problem [32, 33], the Fourier series technique (see Section 6.1.3 and [36, 37]), and variational principles. The periodic eigenstrain method was further developed to determine the overall relaxation moduli of linear viscoelastic composite materials (see Section 7.6 and [38,39]). A particular case, the cell method for periodic media, considers a unit cell with a square inclusion [40].

The analytical procedures mentioned so far yield approximate estimates of the exact solution of the micromechanics problem. These estimates must lie between lower and upper bounds for the solution. Several variational principles were developed to evaluate bounds on the homogenized elastic properties of macroscopically isotropic heterogeneous materials [41]. Those bounds depend only on the volume fractions and the physical properties of the constituents.

In order to study the nonlinear material behavior of composites with periodic microstructure, numerical methods, mainly the finite element method, are employed. Nonlinear finite element analysis of metal matrix composite has been studied by looking at the behavior of the microstructure subjected to an assigned load history [42]. Bounds on overall instantaneous elastoplastic properties of composites have been derived by using the finite element method [43].

6.1 Analytical Homogenization

As discussed in the introduction, estimates of the average properties of heterogeneous media can be obtained by various analytical methods. Detailed derivations of the equations fall outside the scope of this book.

Available analytical models vary greatly in complexity and accuracy. Simple analytical models yield formulas for the stiffness **C** and compliance **S** tensors of the composite [40, (2.9) and (2.12)], such as

$$\begin{aligned} \mathbf{C} &= \sum V_i \ \mathbf{C}^i \mathbf{A}^i \quad ; \quad \sum V_i \mathbf{A}^i = \mathbf{I} \\ \mathbf{S} &= \sum V_i \ \mathbf{S}^i \mathbf{B}^i \quad ; \quad \sum V_i \mathbf{B}^i = \mathbf{I} \end{aligned} \tag{6.1}$$

where $V_i, \mathbf{C}^i, \mathbf{S}^i$ are the volume fraction, stiffness, and compliance tensors (in contracted notation)[1] of the i-th phase in the composite, respectively, and **I** is the 6×6 identity matrix. Furthermore, $\mathbf{A}^i, \mathbf{B}^i$ are the strain and stress concentration tensors (in contracted notation) of the i-th phase [40]. For fiber reinforced composites, $i = f, m$ represent the fiber and matrix phases, respectively.

6.1.1 Reuss Model

The *Reuss model* (also called rule of mixtures) assumes that the strain tensors[2] in the fiber, matrix, and composite are the same $\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^f = \boldsymbol{\varepsilon}^m$; so, the strain concentration tensors are all equal to the 6×6 identity matrix $\mathbf{A}^i = \mathbf{I}$. The rule of mixtures (ROM) formulas for E_1 and ν_{12} are derived and computed in this way.

[1] Fourth-order tensors with minor symmetry are represented by a 6×6 matrix taking advantage of contracted notation.

[2] Tensors are indicated by boldface type, or by their components using index notation.

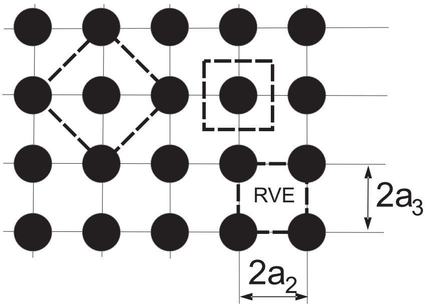


Fig. 6.1: Three possible RVE's for a composite material with a periodic, square fiber array.

6.1.2 Voigt Model

The *Voigt model* (also called inverse rule of mixtures) assumes that the stress tensors in the fiber, matrix, and composite are the same $\boldsymbol{\sigma} = \boldsymbol{\sigma}^f = \boldsymbol{\sigma}^m$; so, the stress concentration tensors are all equal to the 6 × 6 identity matrix $\mathbf{B}^i = \mathbf{I}$. The inverse rule of mixtures (IROM) formulas for E_2 and G_{12} are derived and computed in this way. More realistic concentration tensors are given in [8, Appendix B].

6.1.3 Periodic Microstructure Model

If the composite has periodic microstructure, or if it can be approximated as having such a microstructure (see Section 6.1.4), then Fourier series can be used to estimate all the components of the stiffness tensor of a composite. Explicit formulas for a composite reinforced by isotropic, circular-cylindrical fibers, which are periodically arranged in a square array (Figure 6.1), were developed by [37] and they are presented here. The fibers are aligned with the x_1-axis, and they are equally spaced ($2a_2 = 2a_3$). If the fibers are randomly distributed in the cross section, the resulting composite has transversely isotropic properties, as explained in Section 6.1.4. The case of a composite reinforced with transversely isotropic fibers is presented in [39], and the resulting equations are implemented in [12].

Because the microstructure has square symmetry, the stiffness tensor has six unique coefficients given by

$$C_{11}^* = \lambda_m + 2\,\mu_m - \frac{V_f}{D}\left[\frac{S_3^2}{\mu_m^2} - \frac{2S_6S_3}{\mu_m^2 g} - \frac{aS_3}{\mu_m\,c} + \frac{S_6^2 - S_7^2}{\mu_m^2 g^2} + \frac{aS_6 + bS_7}{\mu_m\,gc} + \frac{a^2 - b^2}{4\,c^2}\right]$$

$$C_{12}^* = \lambda_m + \frac{V_f}{D}b\left[\frac{S_3}{2c\mu_m} - \frac{S_6 - S_7}{2c\mu_m\,g} - \frac{a+b}{4\,c^2}\right]$$

$$C_{23}^* = \lambda_m + \frac{V_f}{D}\left[\frac{aS_7}{2\,\mu_m\,gc} - \frac{ba + b^2}{4\,c^2}\right]$$

$$C_{22}^* = \lambda_m + 2\,\mu_m - \frac{V_f}{D}\left[-\frac{aS_3}{2\,\mu_m\,c} + \frac{aS_6}{2\,\mu_m\,gc} + \frac{a^2 - b^2}{4\,c^2}\right]$$

$$C_{44}^{*} = \mu_m - V_f \left[-\frac{2\,S_3}{\mu_m} + (\mu_m - \mu_f)^{-1} + \frac{4\,S_7}{\mu_m(2-2\nu_m)} \right]^{-1}$$

$$C_{66}^{*} = \mu_m - V_f \left[-\frac{S_3}{\mu_m} + (\mu_m - \mu_f)^{-1} \right]^{-1} \tag{6.2}$$

where

$$\begin{aligned} D = {} & \frac{aS_3^2}{2\,\mu_m^2 c} - \frac{aS_6S_3}{\mu_m^2 gc} + \frac{a(S_6^2 - S_7^2)}{2\mu_m^2 g^2 c} + \\ & + \frac{S_3(b^2-a^2)}{2\,\mu_m\, c^2} + \frac{S_6(a^2-b^2) + S_7(ab+b^2)}{2\,\mu_m g c^2} + \frac{(a^3 - 2b^3 - 3\,ab^2)}{8\,c^3} \end{aligned} \tag{6.3}$$

and

$$\begin{aligned} a &= \mu_f - \mu_m - 2\,\mu_f\,\nu_m + 2\,\mu_m\,\nu_f \\ b &= -\mu_m\,\nu_m + \mu_f\,\nu_f + 2\,\mu_m\,\nu_m\,\nu_f - 2\,\mu_1\,\nu_m\,\nu_f \\ c &= (\mu_m - \mu_f)(\mu_f - \mu_m + \mu_f\,\nu_f - \mu_m\,\nu_m + 2\,\mu_m\,\nu_f - 2\,\mu_f\,\nu_m + \\ & \quad + 2\,\mu_m\,\nu_m\,\nu_f - 2\,\mu_f\,\nu_m\,\nu_f) \\ g &= (2 - 2\nu_m) \end{aligned} \tag{6.4}$$

The subscripts $()_m$, $()_f$ refer to matrix and fiber, respectively. Assuming the fiber and matrix are both isotropic (Section 1.12.5), Lamé constants of both materials are obtained by using (1.75) in terms of the Young's modulus E, the Poisson's ratio ν, and the shear modulus $G = \mu$.

For a composite reinforced by long circular cylindrical fibers, periodically arranged in a square array (Figure 6.1), aligned with x_1-axis, with $a_2 = a_3$, the constants S_3, S_6, S_7 are given as follows [37]

$$\begin{aligned} S_3 &= 0.49247 - 0.47603V_f - 0.02748V_f^2 \\ S_6 &= 0.36844 - 0.14944V_f - 0.27152V_f^2 \\ S_7 &= 0.12346 - 0.32035V_f + 0.23517V_f^2 \end{aligned} \tag{6.5}$$

The resulting tensor $\mathbf{C}^*$ has square symmetry due to the microstructural periodic arrangement in the form of a square array. The tensor $\mathbf{C}^*$ is therefore described by six constants. However, most composites have random arrangement of the fibers (see Figure 1.12), resulting in a transversely isotropic stiffness tensor. A generalization for transversely isotropic materials is presented in Section 6.1.4, next.

6.1.4 Transversely Isotropic Averaging

In order to obtain a transversely isotropic stiffness tensor (Section 1.12.4), equivalent in the average sense to the stiffness tensor with square symmetry, the following averaging procedure is used. A rotation θ of the tensor $\mathbf{C}^*$ about the x_1-axis produces

$$\mathbf{B}(\theta) = \overline{T}^{T}(\theta)\mathbf{C}^{*}\overline{T}(\theta) \tag{6.6}$$

where $\overline{T}(\theta)$ is the coordinate transformation matrix (see (1.50)). Then the equivalent transversely isotropic tensor is obtained by averaging as follows

$$\overline{\mathbf{B}} = \frac{1}{\pi}\int_{0}^{\pi} \mathbf{B}(\theta) d\theta \tag{6.7}$$

Then, using the relations between the engineering constants and the components of the $\overline{\mathbf{B}}$ tensor, the following expressions are obtained explicitly in terms of the coefficients (6.2-6.5) of the tensor $\mathbf{C}^{*}$

$$\begin{aligned}
E_1 &= C_{11}^{*} - \frac{2\,C_{12}^{*2}}{C_{22}^{*} + C_{23}^{*}} \\
E_2 &= \frac{\left(2\,C_{11}^{*}\,C_{22}^{*} + 2\,C_{11}^{*}\,C_{23}^{*} - 4\,C_{12}^{*2}\right)\left(C_{22}^{*} - C_{23}^{*} + 2\,C_{44}^{*}\right)}{3\,C_{11}^{*}\,C_{22}^{*} + C_{11}^{*}\,C_{23}^{*} + 2\,C_{11}^{*}\,C_{44}^{*} - 4\,C_{12}^{*2}} \\
G_{12} &= G_{13} = C_{66}^{*} \\
\nu_{12} &= \nu_{13} = \frac{C_{12}^{*}}{C_{22}^{*} + C_{23}^{*}} \\
\nu_{23} &= \frac{C_{11}^{*}\,C_{22}^{*} + 3\,C_{11}^{*}\,C_{23}^{*} - 2\,C_{11}^{*}\,C_{44}^{*} - 4\,C_{12}^{*2}}{3\,C_{11}^{*}\,C_{22}^{*} + C_{11}^{*}\,C_{23}^{*} + 2\,C_{11}^{*}\,C_{44}^{*} - 4\,C_{12}^{*2}}
\end{aligned} \tag{6.8}$$

Note that the transverse shear modulus G_{23} can be written in terms of the other engineering constants as

$$G_{23} = \frac{C_{22}^{*}}{4} - \frac{C_{23}^{*}}{4} + \frac{C_{44}^{*}}{2} = \frac{E_2}{2(1+\nu_{23})}$$

or directly in terms of μ_m, μ_f as

$$\begin{aligned}
G_{23} = \mu_m - \frac{f}{4D}\,&[(-\frac{aS_3}{2\,\mu_m\,c} + \frac{a(S_7+S_6)}{2\,\mu_m\,gc} - \frac{ba+2b^2-a^2}{4\,c^2}) \\
&+2(-\frac{2\,S_3}{\mu_m} + (\mu_m - \mu_f)^{-1} + \frac{4\,S_7}{\mu_m(2-2\nu_m)})^{-1}]
\end{aligned} \tag{6.9}$$

where D is given by (6.3), $a, b, c,$ and g are given by (6.4), and S_3, S_6, and S_7 can be evaluated by (6.5). These equations are implemented in PMMIE.m and PMMIE.xls [5]. For the case of transversely isotropic fibers, they are implemented in [12].

Example 6.1 *Compute the elastic properties of a composite material reinforced with parallel cylindrical fibers randomly distributed in the cross section. The constituent properties are $E_f = 241\,GPa$, $\nu_f = 0.2$, $E_m = 3.12\,GPa$, $\nu_m = 0.38$, fiber volume fraction $V_f = 0.4$.*

Solution to Example 6.1 *The results shown in Table 6.1 are obtained using [12], which implements the PMM equations for the case of transversely isotropic fibers.*

Table 6.1: Lamina elastic properties for $V_f = 0.4$.

Young's Moduli	Poisson's Ratio	Shear Moduli
$E_1 = 98,306\, MPa$	$\nu_{12} = \nu_{13} = 0.298$	$G_{12} = G_{13} = 2,594\, MPa$
$E_2 = E_3 = 6,552\, MPa$	$\nu_{23} = 0.600$	

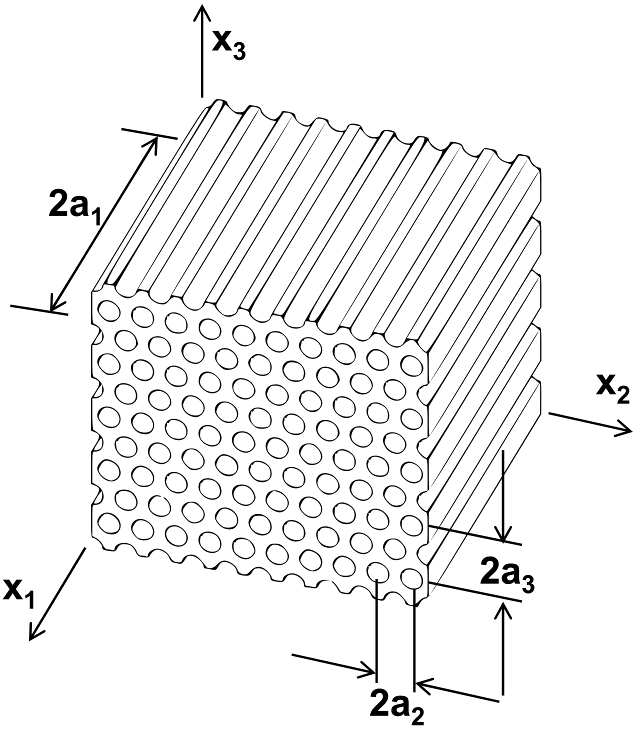


Fig. 6.2: Composite material with hexagonal array.

6.2 Numerical Homogenization

The composite material considered in this section has cylindrical fibers of infinite length, embedded in an elastic matrix, as shown in Figure 6.2. The cross section of the composite obtained by intersecting with a plane orthogonal to the fiber axis is shown in Figure 6.3, which clearly shows a periodic microstructure. Because of the periodicity, the three-dimensional representative volume element (RVE) shown in Figure 6.4 can be used for FE analysis.

In general, composites reinforced with parallel fibers display orthotropic material properties (Section 1.12.3) at the mesoscale (lamina level). In special cases, such as the hexagonal array shown in Figures 6.2 and 6.3, the properties become transversely isotropic (Section 1.12.4). In most commercially fabricated composites, it is impossible to control the placement of the fibers so precisely and most of the time the resulting microstructure is random, as shown in Figure 1.12. A random microstructure results in transversely isotropic properties at the mesoscale. The analysis of composites with random microstructure still can be done using a

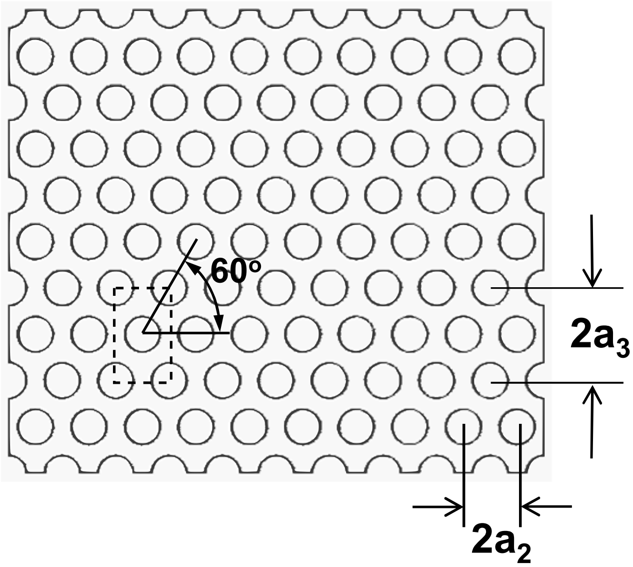


Fig. 6.3: Cross section of the composite material.

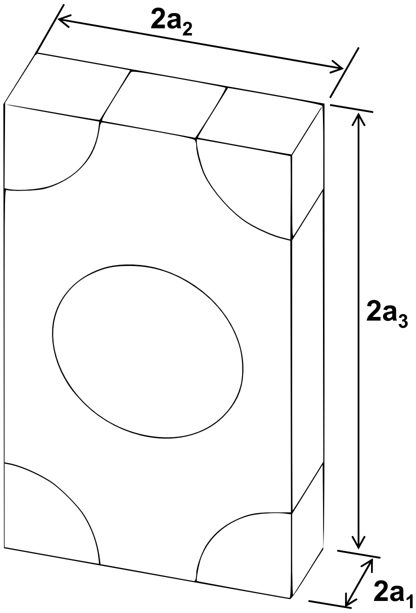


Fig. 6.4: Representative volume element (RVE).

fictitious periodic microstructure, such as that shown in Figure 6.1, then averaging the stiffness tensor **C** as in Section 6.1.4 to obtain the stiffness tensor of a transversely isotropic material. A simpler alternative is to assume that the random microstructure is well approximated by the hexagonal microstructure displayed in Figure 6.3. Analysis of such microstructure directly yields a transversely isotropic stiffness tensor, represented by (1.70), which is reproduced here for convenience

$$\left\{\begin{array}{c}\overline{\sigma}_1\\\overline{\sigma}_2\\\overline{\sigma}_3\\\overline{\sigma}_4\\\overline{\sigma}_5\\\overline{\sigma}_6\end{array}\right\}=\left[\begin{array}{cccccc}C_{11}&C_{12}&C_{12}&0&0&0\\C_{12}&C_{22}&C_{23}&0&0&0\\C_{12}&C_{23}&C_{22}&0&0&0\\0&0&0&\frac{1}{2}(C_{22}-C_{23})&0&0\\0&0&0&0&C_{66}&0\\0&0&0&0&0&C_{66}\end{array}\right]\left\{\begin{array}{c}\overline{\epsilon}_1\\\overline{\epsilon}_2\\\overline{\epsilon}_3\\\overline{\gamma}_4\\\overline{\gamma}_5\\\overline{\gamma}_6\end{array}\right\} \tag{6.10}$$

where the 1-axis aligned with the fiber direction and an over-bar indicates the average computed over the volume of the RVE. Once the components of the transversely isotropic tensor **C** are known, the five elastic properties of the homogenized material can be computed by (6.11), i.e., the longitudinal and transversal Young's moduli E_1 and E_2, the longitudinal and transversal Poisson's ratios ν_{12} and ν_{23}, and the longitudinal shear modulus G_{12}, as follows

$$\begin{aligned}E_1 &= C_{11}-2C_{12}^2/(C_{22}+C_{23})\\ \nu_{12} &= C_{12}/(C_{22}+C_{23})\\ E_2 &= \left[C_{11}\left(C_{22}+C_{23}\right)-2C_{12}^2\right](C_{22}-C_{23})/\left(C_{11}C_{22}-C_{12}^2\right)\\ \nu_{23} &= \left[C_{11}C_{23}-C_{12}^2\right]/\left(C_{11}C_{22}-C_{12}^2\right)\\ G_{12} &= C_{66}\end{aligned} \tag{6.11}$$

The shear modulus G_{23} in the transversal plane can be obtained by the classical relation (1.74) or directly as follows

$$G_{23}=C_{44}=\frac{1}{2}\left(C_{22}-C_{23}\right)=\frac{E_2}{2\left(1+\nu_{23}\right)} \tag{6.12}$$

In order to evaluate the overall elastic matrix **C** of the composite, the RVE is subjected to an average strain $\overline{\epsilon}_\beta$ [44]. The six components of strain ε_{ij}^0 are applied by enforcing the following boundary conditions on the displacement components

$$u_i\left(a_1,x_2,x_3\right)-u_i\left(-a_1,x_2,x_3\right)=2a_1\varepsilon_{i1}^0 \qquad \begin{array}{c}-a_2\le x_2\le a_2\\-a_3\le x_3\le a_3\end{array} \tag{6.13}$$

$$u_i\left(x_1,a_2,x_3\right)-u_i\left(x_1,-a_2,x_3\right)=2a_2\varepsilon_{i2}^0 \qquad \begin{array}{c}-a_1\le x_1\le a_1\\-a_3\le x_3\le a_3\end{array} \tag{6.14}$$

$$u_i\left(x_1,x_2,a_3\right)-u_i\left(x_1,x_2,-a_3\right)=2a_3\varepsilon_{i3}^0 \qquad \begin{array}{c}-a_1\le x_1\le a_1\\-a_2\le x_2\le a_2\end{array} \tag{6.15}$$

Note that tensor components of strain, defined in (1.5), are used in (6.13-6.15). Also, note that a superscript $()^0$ indicates an *applied* strain, while an over-line indicates a volume average. Furthermore, $2a_j\ \varepsilon^0_{ij}$ is the displacement necessary to enforce a strain ε^0_{ij} over a distance $2a_j$ (Figure 6.4).

The strain ε^0_{ij} applied on the boundary by using (6.13-6.15) results in a complex state of strain inside the RVE. However, the volume average of the strain in the RVE equals the applied strain, i.e.,

$$\overline{\varepsilon}_{ij} = \frac{1}{V}\int_V \varepsilon_{ij} dV = \varepsilon^0_{ij} \tag{6.16}$$

For the homogeneous composite material, the relationship between average stress and strain is

$$\overline{\sigma}_\alpha = C_{\alpha\beta}\ \overline{\epsilon}_\beta \tag{6.17}$$

where the relationship between $i, j = 1..3$ and $\beta = 1..6$ is given by the definition of contracted notation in (1.9). Thus, the components of the tensor $\mathbf{C}$ are determined solving six elastic models of the RVE subjected to the boundary conditions (6.13-6.15), where only one component of the strain ϵ^0_β is different from zero for each of the six problems.

By choosing a unit value of applied strain, and once the problem defined by the boundary conditions (6.13-6.15) is solved, it is possible to compute the stress field σ_α, whose average gives the required components of the elastic matrix, one column at a time, as

$$C_{\alpha\beta} = \overline{\sigma}_\alpha = \frac{1}{V}\int_V \sigma_\alpha\left(x_1, x_2, x_3\right)\ dV \quad \text{with } \epsilon^0_\beta = 1 \tag{6.18}$$

where $\alpha, \beta = 1 \ldots 6$ (see Section 1.5). The integrals (6.18) are evaluated within each finite element using the Gauss-Legendre quadrature. Commercial programs, such as *Abaqus*, have the capability to compute the average stress and volume, element by element. Therefore, computation of the integral (6.18) is a trivial matter. For more details see Example 6.2, p. 226.

The coefficients in $\mathbf{C}$ are found by setting a different problem for each column in (6.10), as follows.

First Column of C

In order to determine the components C_{i1}, with $i = 1, 2, 3$, the following strain is applied to stretch the RVE in the fiber direction (x$_1$-direction)

$$\epsilon^o_1 = 1 \qquad \epsilon^o_2 = \epsilon^o_3 = \gamma^o_4 = \gamma^o_5 = \gamma^o_6 = 0 \tag{6.19}$$

Thus, the displacement boundary conditions (6.13-6.15) for the RVE in Figure 6.4 become

$$\begin{aligned}
&\left.\begin{aligned}
u_1\left(+a_1, x_2, x_3\right)-u_1\left(-a_1, x_2, x_3\right) &= 2a_1\\
u_2\left(+a_1, x_2, x_3\right)-u_2\left(-a_1, x_2, x_3\right) &= 0\\
u_3\left(+a_1, x_2, x_3\right)-u_3\left(-a_1, x_2, x_3\right) &= 0
\end{aligned}\right. \qquad \begin{aligned}-a_2\le x_2\le a_2\\ -a_3\le x_3\le a_3\end{aligned}\\
&u_i\left(x_1, +a_2, x_3\right)-u_i\left(x_1, -a_2, x_3\right) = 0 \qquad \begin{aligned}-a_1\le x_1\le a_1\\ -a_3\le x_3\le a_3\end{aligned}\\
&u_i\left(x_1, x_2, +a_3\right)-u_i\left(x_1, x_2, -a_3\right) = 0 \qquad \begin{aligned}-a_1\le x_1\le a_1\\ -a_2\le x_2\le a_2\end{aligned}
\end{aligned} \tag{6.20}$$

The conditions (6.20) are constraints on the relative displacements between opposite faces of the RVE. Because of the symmetries of the RVE and symmetry of the constraints (6.20), only one-eighth of the RVE needs to be modeled in FEA. Assuming the top-right-front portion is modeled (Figure 6.5), the following equivalent external boundary conditions, i.e., boundary conditions on components of displacements and stresses, can be used

$$\begin{aligned}
&\begin{aligned}
u_1\left(a_1, x_2, x_3\right) &= a_1\\
u_1\left(0, x_2, x_3\right) &= 0\\
\sigma_{12}\left(a_1, x_2, x_3\right) &= 0\\
\sigma_{12}\left(0, x_2, x_3\right) &= 0\\
\sigma_{13}\left(a_1, x_2, x_3\right) &= 0\\
\sigma_{13}\left(0, x_2, x_3\right) &= 0
\end{aligned} \qquad \begin{aligned}0\le x_2\le a_2\\ 0\le x_3\le a_3\end{aligned}\\
&\begin{aligned}
u_2\left(x_1, a_2, x_3\right) &= 0\\
u_2\left(x_1, 0, x_3\right) &= 0\\
\sigma_{21}\left(x_1, a_2, x_3\right) &= 0\\
\sigma_{21}\left(x_1, 0, x_3\right) &= 0\\
\sigma_{23}\left(x_1, a_2, x_3\right) &= 0\\
\sigma_{23}\left(x_1, 0, x_3\right) &= 0
\end{aligned} \qquad \begin{aligned}0\le x_1\le a_1\\ 0\le x_3\le a_3\end{aligned}\\
&\begin{aligned}
u_3\left(x_1, x_2, a_3\right) &= 0\\
u_3\left(x_1, x_2, 0\right) &= 0\\
\sigma_{31}\left(x_1, x_2, a_3\right) &= 0\\
\sigma_{31}\left(x_1, x_2, 0\right) &= 0\\
\sigma_{32}\left(x_1, x_2, a_3\right) &= 0\\
\sigma_{32}\left(x_1, x_2, 0\right) &= 0
\end{aligned} \qquad \begin{aligned}0\le x_1\le a_1\\ 0\le x_2\le a_2\end{aligned}
\end{aligned} \tag{6.21}$$

These boundary conditions are very easy to apply. Symmetry boundary conditions are applied on the planes $x_1 = 0$, $x_2 = 0$, $x_3 = 0$. Then, a uniform displacement is applied on the plane $x_1 = a_1$. The stress boundary conditions do not need to be applied explicitly in a displacement-based formulation. The displacement components in (6.21) represent strains that are non-zero along the x_1-direction and zero along the other two directions. The stress boundary conditions listed in (6.21) reflect the fact that, in the coordinate system used, the composite material

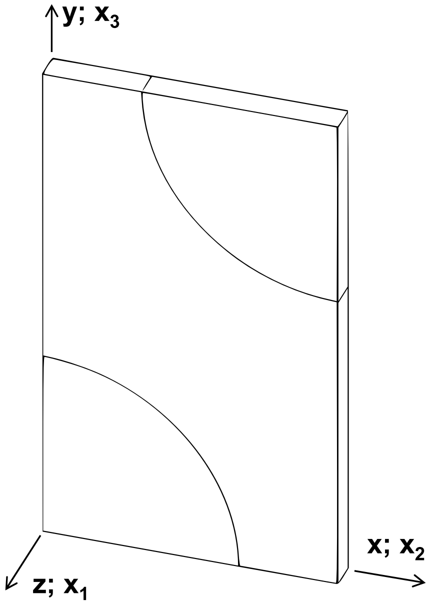


Fig. 6.5: One-eighth model of the RVE. Note that the model is set up with the fiber along the z-axis, which corresponds to the x_1-direction in the equations.

is macroscopically orthotropic and that the constituent materials are orthotropic too. Therefore, there is no coupling between extension and shear strains. This is evidenced by the zero coefficients above the diagonal in columns 4 to 6 in (6.10).

The coefficients in column one of (6.10) are found by using (6.18), as follows

$$C_{\alpha 1} = \overline{\sigma}_\alpha = \frac{1}{V}\int_V \sigma_\alpha\left(x_1, x_2, x_3\right)\ dV \tag{6.22}$$

Second Column of C

The components $C_{\alpha 2}$, with $\alpha = 1, 2, 3$, are determined by setting

$$\epsilon_2^o = 1 \qquad \epsilon_1^o = \epsilon_3^o = \gamma_4^o = \gamma_5^o = \gamma_6^o = 0 \tag{6.23}$$

Thus, the following boundary conditions on displacements can be used

$$\begin{array}{lcl} u_1\left(a_1, x_2, x_3\right) & = & 0 \\ u_1\left(0, x_2, x_3\right) & = & 0 \\ u_2\left(x_1, a_2, x_3\right) & = & a_2 \\ u_2\left(x_1, 0, x_3\right) & = & 0 \\ u_3\left(x_1, x_2, a_3\right) & = & 0 \\ u_3\left(x_1, x_2, 0\right) & = & 0 \end{array} \tag{6.24}$$

The trivial stress boundary conditions have not been listed because they are automatically enforced by the displacement-based FEA formulation. Using (6.18), the stiffness terms in the second column of **C** are computed as

$$C_{\alpha 2} = \overline{\sigma}_{\alpha} = \frac{1}{V} \int_V \sigma_{\alpha 2}(x_1, x_2, x_3)\ dV \tag{6.25}$$

Third Column of C

Because of the transverse isotropy of the material (6.10), the components of the third column of the matrix **C** can be determined from the first and the second column, so no further computation is required. However, if desired, the components $C_{\alpha 3}$, with $\alpha = 1, 2, 3$, can be found by applying the following strain

$$\epsilon_3^o = 1 \qquad \epsilon_1^o = \epsilon_2^o = \gamma_4^o = \gamma_4^o = \gamma_5^o = 0 \tag{6.26}$$

Thus, the following boundary conditions on displacement can be used

$$\begin{array}{lcl} u_1(a_1, x_2, x_3) & = & 0 \\ u_1(0, x_2, x_3) & = & 0 \\ u_2(x_1, a_2, x_3) & = & 0 \\ u_2(x_1, 0, x_3) & = & 0 \\ u_3(x_1, x_2, a_3) & = & a_3 \\ u_3(x_1, x_2, 0) & = & 0 \end{array} \tag{6.27}$$

The required components of **C** are determined by averaging the stress field as in (6.18).

Example 6.2 *Compute E_1, E_2, ν_{12}, and ν_{23} for a unidirectional composite with isotropic fibers $E_f = 241$ GPa, $\nu_f = 0.2$, and isotropic matrix $E_m = 3.12$ GPa, $\nu_m = 0.38$ with fiber volume fraction $V_f = 0.4$. The fiber diameter is $d_f = 7$ μm, placed in an hexagonal array as shown in Figure 6.3.*

Solution to Example 6.2 *The dimensions a_2 and a_3 of the RVE, as shown in Figure 6.4, are chosen to obtain $V_f = 0.4$ with an hexagonal array microstructure. The fiber volume and the total volume of the RVE are*

$$v_f = 4a_1\pi\left(\frac{d_f}{2}\right)^2 \quad ; \quad v_t = 2a_1\ 2a_2\ 2a_3$$

The ratio between both is the volume fraction. Therefore,

$$V_f = \pi\frac{(d_f/2)^2}{2\ a_2\ a_3} = 0.4$$

Additionally, the relation between a_2 and a_3 is established by the hexagonal array pattern

$$a_3 = a_2 \tan(60°)$$

These two relations yield a_2 and a_3, while the a_1 dimension can be chosen arbitrarily. In this case, the RVE dimensions are

$$a_1 = a_2/4 = 1.3175\ \mu m \quad ; \quad a_2 = 5.270\ \mu m \quad ; \quad a_3 = 9.128\ \mu m$$

Equations (6.18) will be used to obtain the coefficients $C_{11}, C_{12}, C_{13}, C_{22}, C_{23}$, and C_{33} by applying unit strains $\epsilon_1^0, \epsilon_2^0, \epsilon_3^0$, one at at a time. In the absence of applied shear deformations, *the average deformations are symmetric with respect to the three coordinate planes. The RVE in Figure 6.5 is also geometrically symmetric with respect to the three coordinate planes. Therefore, it is possible to model one-eighth of the RVE, as shown in Figure 6.5. The procedure used to model one-eighth of the RVE in* Abaqus *is given below.*

i. Setting the `Work Directory`

```
Menu: File, Set Work Directory, [C:\SIMULIA\user\Ex_6.2]
Menu: File, Save As, [C:\SIMULIA\user\Ex_6.2\Ex_6.2.cae]
```

ii. Creating parts

```
# Values used below, replace the variables accordingly.
# values in microns
# a_1 = 1.3175        # half RVE along x_1 (extrusion direction)
# a_2 = 5.270         # half RVE along x_2
# a_3 = 9.138         # half RVE along x_3
# r_f = 3.5           # fiber radius
Module: Part
Menu:   Part, Create
    3D, Deformable, Solid, Extrusion, Cont
Menu:   Add, Line, Rectangle, [0,0], [a_2, a_3], X, Done
    Depth [a_1], OK
Menu:   Tools, Partition
    Cell, Sketch planar partition, Sketch Origin: Specify
    # pick: front face (z=a_1), Sketch origin X,Y,Z: [0,0,a_1]
    Select and edge: vertical and on the right, # pick: right vertical
Menu:   Add, Circle
    [0,0], [0,r_f]
    [a_2,a_3], [8.77,a_3], X, Done
    Extrude/Sweep, # pick: lines forming lower quarter-circle, Done
    Extrude Along Direction, # pick edge parallel to Z-global, OK
    Create Partition
    # pick: top cell (larger cell), Done
    # pick: the lines forming the upper quarter-circle, Done
    Extrude Along Direction, # pick edge parallel to Z-global, OK
    Create Partition, Done, # close Create Partition pop-up window
```

iii. Defining materials, sections, and assigning sections to parts

```
Module: Property
Menu:   Material, Create
    Name [Fiber], Mechanical, Elasticity, Elastic, Type: Isotropic
    [241000, 0.2], OK
Menu:   Material, Create
    Name [Matrix], Mechanical, Elasticity, Elastic, Type: Isotropic
    [3120, 0.38], OK
Menu:   Section, Create
    Name [Fiber], Solid, Homogeneous, Cont, Material: Fiber, OK
Menu:   Section, Create
    Name [Matrix], Solid, Homogeneous, Cont, Material: Matrix, OK
Menu:   Assign, Section
    # pick: the matrix, Done, Section: Matrix, OK
    # pick: both fibers, Done, Section: Fiber, OK, Done
```

iv. *Creating assembly*

```
Module: Assembly
Menu:   Instance, Create, Independent, OK
```

v. *Defining analysis steps. We need three steps, one for each column in Section 6.2. To solve this example, it will be necessary to include output variables* S *and* IVOL *on the Output Requests.*

```
Module: Step
Menu:   Step, Manager
    Create, Name [Column-1], Insert new step after: Initial
    Procedure type: Linear perturbation, Static, Cont, OK
    Create, Name [Column-2], Insert new step after: Column-1
    Procedure type: Linear perturbation, Static, Cont, OK
    Create, Name [Column-3], Insert new step after: Column-2
    Procedure type: Linear perturbation, Static, Cont, OK
    # close Step Manager pop-up window
Step:   Column-1
Menu:   Output, Field Output Requests, Edit, F-Output-1
    Edit variables, [S,E,U,IVOL], OK
```

vi. *Adding loads and BC.*
The boundary conditions are defined in three load steps, which are then used to obtain the coefficients $C_{\alpha\beta}$ in columns one, two, and three. A unit strain is applied along each direction, each time. Equations (6.18) are then used to obtain the stiffness coefficients. The procedure is shown below.

```
Module: Load
    # each step has its own set of BC
    # for step: Column-1
Menu:   BC, Manager
    Create, Name [xsymm-C1], Step: Column-1, Mechanical,
        Symm/Anti/Enca, Cont
    # pick: faces @x=0 and a_2, Done, XSYMM, OK
    Create, Name [ysymm-C1], Step: Column-1, Mechanical,
        Symm/Anti/Enca, Cont
    # pick: faces @y=0 and a_3, Done, YSYMM, OK
    Create, Name [zsymm-C1], Step: Column-1, Mechanical,
        Symm/Anti/Enca, Cont
    # pick: face @z=0, Done, ZSYMM, OK, # note only one face is picked
    # imposed displacement(unit strain)
    Create, Name [disp-C1], Step: Column-1, Mech., Disp/Rota, Cont
    # pick: faces @z=a_1, Done, # Checkmark: U3 [a_1], OK
    # for step: Column-2
    Create, Name [zsymm-C2], Step: Column-2, Mechanical,
        Symm/Anti/Enca, Cont
    # pick: faces @z=0, and a_1, Done, ZSYMM, OK
    Create, Name [ysymm-C2], Step: Column-2, Mechanical,
        Symm/Anti/Enca, Cont
    # pick: faces @y=0 and a_3, Done, YSYMM, OK
```

```
        Create, Name [xsymm-C2], Step: Column-2, Mechanical,
            Symm/Anti/Enca, Cont
        # pick: face @x=0, Done, XSYMM, OK, # note only one face is picked
        # imposed displacement(unit strain)
        Create, Name [disp-C2], Step: Column-2, Mech., Disp/Rota, Cont
        # pick: face @x=a_2, Done, # Checkmark: U1 [a_2], OK
        # for step: Column-3
        Create, Name [xsymm-C3], Step: Column-3, Mechanical,
            Symm/Anti/Enca, Cont
        # pick: faces @x=0, and a_2, Done, XSYMM, OK
        Create, Name [zsymm-C3], Step: Column-3, Mechanical,
            Symm/Anti/Enca, Cont
        # pick: faces @z=0, and a_1, Done, ZSYMM, OK
        Create, Name [ysymm-C3], Step: Column-3, Mechanical,
            Symm/Anti/Enca, Cont
        # pick: face @y=0, Done, YSYMM, OK, # note only one face is picked
        # imposed displacement(unit strain)
        Create, Name [disp-C3], Step: Column-3, Mech., Disp/Rota, Cont
        # pick: face @y=a_3, Done, # Checkmark: U2 [a_3], OK
        # close the BC Manager pop-up window
```

vii. *Meshing the model*

```
    Module: Mesh
    Menu:   Seed, Instance, Approximate global size [0.5], OK
    Menu:   Mesh, Controls, # select all cells, Done
        Element Shape: Hex, Technique: Structured, OK
    Menu:   Mesh, Element Type, # select all cells, Done
        Geometric Order: Linear, # checkmark: Reduced integration, OK
    Menu:   Mesh, Instance, Yes
```

viii. *Solving and visualizing the results*

```
    Module: Job
    Menu:   Job, Manager
        Create, Cont, OK, Submit, # when Completed, Results
```

The coefficients $C_{\alpha\beta}$ and the laminate elastic constants are computed using the Python script given below, which can be executed by typing or pasting in the Python shell. Alternatively, one can execute any Python script by doing `Menu: File, Run Script`*. The script* `srecover.py` *is available in [5] and it is used in this example. Now, click* `>>>` *to switch from the Information window to the Python shell. Then, type or copy the lines below to the Python shell.*

```
# srecover.py
from visualization import *
# Open the Output Data Base for the current Job
odb = openOdb(path='Job-1.odb');
myAssembly = odb.rootAssembly;
#
# Creating a temporary variable to hold the frame repository
# provides the same functionality and speeds up the process
# Column-1
frameRepository = odb.steps['Column-1'].frames;
frameS=[];
frameIVOL=[];
# Get only the last frame [-1]
frameS.insert(0,frameRepository[-1].fieldOutputs['S']
    .getSubset(position=INTEGRATION_POINT));
frameIVOL.insert(0,frameRepository[-1].fieldOutputs['IVOL']
    .getSubset(position=INTEGRATION_POINT));
# Total Volume
Tot_Vol=0;
# Stress Sum
Tot_Stress=0;
#
for II in range(0,len(frameS[-1].values)):
     Tot_Vol=Tot_Vol+frameIVOL[0].values[II].data;
     Tot_Stress=Tot_Stress+frameS[0].values[II].data * frameIVOL[0]\
        .values[II].data;

# Calculate Average
Avg_Stress = Tot_Stress/Tot_Vol;
print 'Abaqus/Standard Stress Tensor Order:'
# from Abaqus Analysis User's Manual - 1.2.2 Conventions -
# Convention used for stress and strain components
print 'Average stresses Global CSYS: 11-22-33-12-13-23';
print Avg_Stress;
C11 = Avg_Stress[2]#z-component,1-direction
C21 = Avg_Stress[0]#x-component,2-direction
C31 = Avg_Stress[1]#y-component,3-direction in Fig. 6.5

# Column-2
frameRepository = odb.steps['Column-2'].frames;
frameS=[];
frameIVOL=[];
# Get only the last frame [-1]
frameS.insert(0,frameRepository[-1].fieldOutputs['S']
    .getSubset(position=INTEGRATION_POINT));
frameIVOL.insert(0,frameRepository[-1].fieldOutputs['IVOL']
    .getSubset(position=INTEGRATION_POINT));
# Total Volume
Tot_Vol=0;
# Stress Sum
Tot_Stress=0;
#
for II in range(0,len(frameS[-1].values)):
     Tot_Vol=Tot_Vol+frameIVOL[0].values[II].data;
     Tot_Stress=Tot_Stress+frameS[0].values[II].data * frameIVOL[0]\
        .values[II].data;

# Calculate Average
Avg_Stress = Tot_Stress/Tot_Vol;
```

```
print 'Abaqus/Standard Stress Tensor Order:'
print 'Average stresses Global CSYS: 11-22-33-12-13-23';
print Avg_Stress;
C12 = Avg_Stress[2]#z-component,1-direction
C22 = Avg_Stress[0]#x-component,2-direction
C32 = Avg_Stress[1]#y-component,3-direction in Fig. 6.5

# Column-3
frameRepository = odb.steps['Column-3'].frames;
frameS=[];
frameIVOL=[];
# Get only the last frame [-1]
frameS.insert(0,frameRepository[-1].fieldOutputs['S']
    .getSubset(position=INTEGRATION_POINT));
frameIVOL.insert(0,frameRepository[-1].fieldOutputs['IVOL']
    .getSubset(position=INTEGRATION_POINT));
# Total Volume
Tot_Vol=0;
# Stress Sum
Tot_Stress=0;
#
for II in range(0,len(frameS[-1].values)):
     Tot_Vol=Tot_Vol+frameIVOL[0].values[II].data;
     Tot_Stress=Tot_Stress+frameS[0].values[II].data * frameIVOL[0]\
        .values[II].data;

# Calculate Average
Avg_Stress = Tot_Stress/Tot_Vol;
print 'Abaqus/Standard Stress Tensor Order:'
print 'Average stresses in Global CSYS: 11-22-33-12-13-23';
print Avg_Stress;
C13 = Avg_Stress[2]#z-component,1-direction
C23 = Avg_Stress[0]#x-component,2-direction
C33 = Avg_Stress[1]#y-component,3-direction in Fig. 6.5
#
EL=C11-2*C12*C21/(C22+C23)              # Longitudinal E1 modulus
nuL=C12/(C22+C23)                       # 12 Poisson coefficient
ET=(C11*(C22+C23)-2*C12*C12)*(C22-C23)/(C11*C22-C12*C21)    #E2
nuT=(C11*C23-C12*C21)/(C11*C22-C12*C21) # 23 Poisson coefficient
GT=(C22-C23)/2 # or GT=ET/2/(1+nuT)     # 23 Shear stiffness
#
print "If Moduli are in TPa and dimensions in microns, results are in TPa"
print "E1=",EL
print "E2=",ET
print "PR12=",nuL
print "PR23=",nuT
# end srecover.py
```

The results are shown in Table 6.2. If you get any errors, check that the blank lines following each of the `for` loops are completely empty. These blank lines are needed as terminators for the `for` loops in *Python*. Otherwise, try running the

Table 6.2: Calculated elastic properties of the unidirectional lamina.

Property	PMM	FEA
E_1 [MPa]	98,306	98,197
E_2 [MPa]	6,552	7,472
ν_{12}	0.298	0.299
ν_{23}	0.6	0.540
G_{12}	2,594	(*)

(*) Not possible with the boundary conditions used in this example.

script as follows: `Menu: File, Run Script: [srecover.py], OK`.

The modeling procedure for Example 6.2 is quite involved. If one wishes to perform a parametric study, repeating this process on an interactive session using the CAE graphical user interface (GUI) would be very time consuming and prone to error. Instead, it is possible to capture the Python script generated by CAE during an interactive session and use it to automate the process.

Fourth Column of C

For a transversally isotropic material, according to (6.10), only the term C_{44} is expected to be different from zero and it can be determined as a function of the other components, so no further computation is needed. Therefore, it can be determined as

$$C_{44} = \frac{1}{2}(C_{22} - C_{23}) \tag{6.28}$$

If the material is orthotropic, a procedure similar to that used for column number six must be used.

Fifth Column of C

For a transversally isotropic material, according to (6.10), only the term C_{55} is different from zero and it is equal to C_{66}, which can be found from column number six. If the material is orthotropic, a procedure similar to that used for column number six must be used.

Sixth Column of C

Because of the lack of symmetry of the loads, in this case it is not possible to use boundary conditions as was done for the first three columns. Thus, the boundary conditions must be enforced by using coupling constraint equations (called CE in most FEA commercial packages).

According to (6.10), only the term C_{66} is different from zero. The components $C_{\alpha 6}$ are determined by setting

$$\gamma_6^0 = \varepsilon_{12}^0 + \varepsilon_{21}^0 = 1.0 \qquad \epsilon_1^0 = \epsilon_2^0 = \epsilon_3^0 = \gamma_4^0 = \gamma_5^0 = 0 \tag{6.29}$$

Note that $\varepsilon_{12}^0 = 1/2$ is applied between $x_1 = \pm a_1$ and another one-half is applied between $x_2 = \pm a_2$. In this case, the CE applied between two periodic faces (except points in the edges and vertices) is given as a particular case of (6.13-6.15) as follows

$$\begin{array}{rcll}
u_1(a_1,x_2,x_3) - u_1(-a_1,x_2,x_3) &=& 0 & \\
u_2(a_1,x_2,x_3) - u_2(-a_1,x_2,x_3) &=& a_1 & -a_2 < x_2 < a_2 \\
u_3(a_1,x_2,x_3) - u_3(-a_1,x_2,x_3) &=& 0 & -a_3 < x_3 < a_3 \\
u_1(x_1,a_2,x_3) - u_1(x_1,-a_2,x_3) &=& a_2 & \\
u_2(x_1,a_2,x_3) - u_2(x_1,-a_2,x_3) &=& 0 & -a_1 < x_1 < a_1 \\
u_3(x_1,a_2,x_3) - u_3(x_1,-a_2,x_3) &=& 0 & -a_3 < x_3 < a_3 \\
u_1(x_1,x_2,a_3) - u_1(x_1,x_2,-a_3) &=& 0 & \\
u_2(x_1,x_2,a_3) - u_2(x_1,x_2,-a_3) &=& 0 & -a_1 < x_1 < a_1 \\
u_3(x_1,x_2,a_3) - u_3(x_1,x_2,-a_3) &=& 0 & -a_2 < x_2 < a_2
\end{array} \tag{6.30}$$

Note that (6.30) are applied between opposite points on the faces of the RVE but not on edges and vertices. In FEA, CE are applied between degrees of freedom (DOF). Once a DOF has been used in a CE, it cannot be used in another CE. For example, the first of (6.30) for $x_2 = a_2$ becomes

$$u_1(a_1,a_2,x_3) - u_1(-a_1,a_2,x_3) = 0 \tag{6.31}$$

The DOF associated to $u_1(a_1,a_2,x_3)$ (for all $-a_3 < x_3 < a_3$) are eliminated because they are identical to $u_1(-a_1,a_2,x_3)$, as required by (6.31) and enforced by a CE based on the same. Once the DOF are eliminated, they cannot be used in another CE. For example, the fourth of (6.30) at $x_1 = a_1$ is

$$u_1(a_1,a_2,x_3) - u_1(a_1,-a_2,x_3) = 0 \tag{6.32}$$

but this CE cannot be enforced because the DOF associated to $u_1(a_1,a_2,x_3)$ have been eliminated by the CE associated to (6.31). As a corollary, constraint equations on the edges and vertices of the RVE must be written separately from (6.30). Furthermore, only three equations, one for each component of displacement u_i, can be written between a pair of edges or pair of vertices. Simply put, there are only three displacements that can be used to enforce periodicity conditions.

For pairs of edges, the task at hand is to reduce the first six equations of (6.30) to three equations that can be applied between pairs of edges for the interval $-a_3 < x_3 < a_3$. Note that the new equations will not be applied at $x_3 = \pm a_3$ because those are vertices, which will be dealt with separately. Therefore, the last three equations of (6.30) are inconsequential at this point.

The only way to reduce six equations to three, in terms of six unique DOF, is to add the equations for diagonally opposite edges. Figure 6.6 is a top view of the RVE looking from the positive x_3-axis. Point A in Figure 6.6 represents the edge formed by the planes $x_1 = a_1$ and $x_2 = a_2$. This location is constrained by the first of (6.30) at that location, which is precisely (6.31). Point C in Figure 6.6 represents

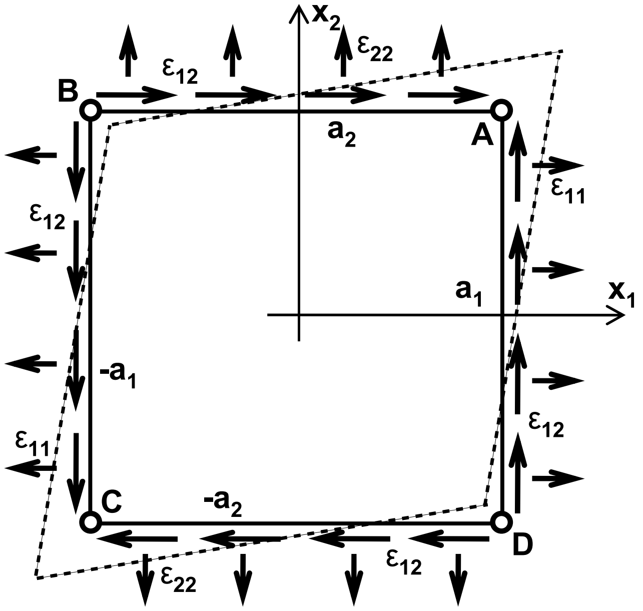


Fig. 6.6: Top view of the RVE showing that two displacements (vertical and horizontal) must be applied at edges to impose shear strain (shown as points A, B, C, and D in the figure).

the edge formed by the planes $x_1 = -a_1$ and $x_2 = -a_2$. This location is constrained by the fourth of (6.30), which at that location reduces to

$$u_1(-a_1, a_2, x_3) - u_1(-a_1, -a_2, x_3) = a_2 \tag{6.33}$$

Adding (6.31) and (6.33) yields a single equation as follows

$$u_1(a_1, a_2, x_3) - u_1(-a_1, -a_2, x_3) = a_2 \tag{6.34}$$

Repeating the procedure for the components u_2 and u_3, and grouping the resulting equations with (6.34), results in

$$\begin{aligned} u_1(a_1, a_2, x_3) - u_1(-a_1, -a_2, x_3) &= a_2 \\ u_2(a_1, a_2, x_3) - u_2(-a_1, -a_2, x_3) &= a_1 \qquad -a_3 < x_3 < a_3 \\ u_3(a_1, a_2, x_3) - u_3(-a_1, -a_2, x_3) &= 0 \end{aligned} \tag{6.35}$$

Considering (6.30) between edges B and D in Figure 6.6 results in

$$\begin{aligned} u_1(a_1, -a_2, x_3) - u_1(-a_1, a_2, x_3) &= -a_2 \\ u_2(a_1, -a_2, x_3) - u_2(-a_1, a_2, x_3) &= a_1 \qquad -a_3 < x_3 < a_3 \\ u_3(a_1, -a_2, x_3) - u_3(-a_1, a_2, x_3) &= 0 \end{aligned} \tag{6.36}$$

The planes $x_1 = \pm a_1$ and $x_3 = \pm a_3$ define two pairs of edges restrained by the following six CE

$$
\begin{array}{rcll}
u_1\left(+a_1,x_2,+a_3\right)-u_1\left(-a_1,x_2,-a_3\right) & = & 0 & \\
u_2\left(+a_1,x_2,+a_3\right)-u_2\left(-a_1,x_2,-a_3\right) & = & a_1 & \quad -a_2<x_2<a_2 \\
u_3\left(+a_1,x_2,+a_3\right)-u_3\left(-a_1,x_2,-a_3\right) & = & 0 & \\
u_1\left(+a_1,x_2,-a_3\right)-u_1\left(-a_1,x_2,+a_3\right) & = & 0 & \\
u_2\left(+a_1,x_2,-a_3\right)-u_2\left(-a_1,x_2,+a_3\right) & = & a_1 & \quad -a_2<x_2<a_2 \\
u_3\left(+a_1,x_2,-a_3\right)-u_3\left(-a_1,x_2,+a_3\right) & = & 0 &
\end{array}
\tag{6.37}
$$

The six CE for the two pairs of edges defined by the planes $x_2 = \pm a_2$ and $x_3 = \pm a_3$ are

$$
\begin{array}{rcll}
u_1\left(x_1,+a_2,+a_3\right)-u_1\left(x_1,-a_2,-a_3\right) & = & a_2 & \\
u_2\left(x_1,+a_2,+a_3\right)-u_2\left(x_1,-a_2,-a_3\right) & = & 0 & \quad -a_1<x_1<a_1 \\
u_3\left(x_1,+a_2,+a_3\right)-u_3\left(x_1,-a_2,-a_3\right) & = & 0 & \\
u_1\left(x_1,+a_2,-a_3\right)-u_1\left(x_1,-a_2,+a_3\right) & = & a_2 & \\
u_2\left(x_1,+a_2,-a_3\right)-u_2\left(x_1,-a_2,+a_3\right) & = & 0 & \quad -a_1<x_1<a_1 \\
u_3\left(x_1,+a_2,-a_3\right)-u_3\left(x_1,-a_2,+a_3\right) & = & 0 &
\end{array}
\tag{6.38}
$$

Note that (6.35-6.38) are not applied at the vertices because redundant CE would appear among pairs of vertices that are located symmetrically with respect to the center of the RVE's volume. Therefore, each of the four pairs of vertices need to be constrained one at a time. The resulting CE are as follows

$$
\begin{array}{rl}
u_1(+a_1,+a_2,+a_3)-u_1(-a_1,-a_2,-a_3) & =a_2 \\
u_2(+a_1,+a_2,+a_3)-u_2(-a_1,-a_2,-a_3) & =a_1 \\
u_3(+a_1,+a_2,+a_3)-u_3(-a_1,-a_2,-a_3) & =0 \\
u_1(+a_1,+a_2,-a_3)-u_1(-a_1,-a_2,+a_3) & =a_2 \\
u_2(+a_1,+a_2,-a_3)-u_2(-a_1,-a_2,+a_3) & =a_1 \\
u_3(+a_1,+a_2,-a_3)-u_3(-a_1,-a_2,+a_3) & =0 \\
u_1(-a_1,+a_2,+a_3)-u_1(+a_1,-a_2,-a_3) & =a_2 \\
u_2(-a_1,+a_2,+a_3)-u_2(+a_1,-a_2,-a_3) & =-a_1 \\
u_3(-a_1,+a_2,+a_3)-u_3(+a_1,-a_2,-a_3) & =0 \\
u_1(+a_1,-a_2,+a_3)-u_1(-a_1,+a_2,-a_3) & =-a_2 \\
u_2(+a_1,-a_2,+a_3)-u_2(-a_1,+a_2,-a_3) & =a_1 \\
u_3(+a_1,-a_2,+a_3)-u_3(-a_1,+a_2,-a_3) & =0
\end{array}
\tag{6.39}
$$

Equations (6.30-6.39) constrain the volume of the RVE with a unit strain given by (6.29). The FEA of this model yields all the components of stress. As discussed previously, element by element averages of these components of stress are available from the FEA (see macro `srecover` in Example 6.1, p. 219) or they can be easily computed by post-processing. Therefore, the coefficient C_{66}, for this case is found using (6.18) written as

$$C_{66} = \overline{\sigma}_6 = \frac{1}{V}\int_V \sigma_6\left(x_1, x_2, x_3\right)\, dV \quad \text{with } \gamma_6^0 = 1 \tag{6.40}$$

Finally, the elastic properties of the composite are determined using (6.11).

Example 6.3 *Compute G_{12} for the composite in Example 6.2 (p. 226).*

Solution to Example 6.3 *First note that for the composite reinforced with an hexagonal array of fibers, $G_{12} = G_{13}$. Therefore, to compute $G_{13} = C_{55}$ we can apply a deformation $\gamma_5^0 = 2\epsilon_{13}^0 = 1.0$. Such state of average deformation is antisymmetric with respect to the (x_1, x_2) coordinate plane (X, Z plane in the* Abaqus *model shown in Figure 6.7) and symmetric with respect to the (x_1, x_3) plane (Y, Z plane in the model). Therefore, we choose to model the full RVE. The deformed shape and contour plot of strain γ_{13} (E23 in the model) is shown in Figure 6.7. Such plot corresponds to an average deformation $\epsilon_{13} = 1.0$ and average stress $\sigma_{13} = 2,579\ MPa$, thus yielding $G_{13} = 2,579\ MPa$ according to*

$$C_{55} = \overline{\sigma}_5 = \frac{1}{V}\int_V \sigma_5\left(x_1, x_2, x_3\right)\, dV \quad \textit{with } \gamma_5^0 = 1 \tag{6.41}$$

The fiber diameter is $d_f = 7\ \mu m$ and the RVE dimensions are:

$$a_1 = a_2/4 = 1.3175\ \mu m \quad ; \qquad a_2 = 5.270\ \mu m \quad ; \qquad a_3 = 9.128\ \mu m$$

The geometry (`Module: Part`), material and sections (`Module: Property`), assembly (`Module: Assembly`), step (`Module: Step`), and mesh (`Module: Mesh`) can be created in the same fashion as it was done in Example 6.2, p. 226.

Model coordinates X, Y, Z (see Figure 6.7) are used in this section to describe the boundary conditions. The bottom of the RVE, at $Y = -a_3$ in the model, is fixed. The top is subjected to a displacement $U3 = 2 \times a_3$ to represent the applied shear strain $\epsilon_{13} = 1.0$. The sides of the RVE, at $X = \pm a_2$, are subject to symmetry BC, which emulate the periodicity conditions in this case because the deformation is symmetric with respect to the plane Y, Z in the model. The pseudo code below describes how to apply the boundary conditions.

```
Module: Load
Menu:   BC, Manager
    Create, Name [ysupp], Step: Step-1, Mechanical, Disp/Rota, Cont
    # pick the face @ y=-a_3, Done, # Checkmark: U2 [0] and U3 [0], OK
    Create, Name [ydisp], Step: Step-1, Mechanical, Disp/Rota, Cont
    # pick the face @ y=a_3, Done, # Checkmark: U3 [18.276] # 2*a_3, OK
    Create, Name [xsymm], Step: Step-1, Mechanical, Symm/Anti/Enca, Cont
    # pick the faces @ x=-a_2 and x=a_2, Done, XSYMM, OK
    # close BC Manager pop-up window
```

Enforcing periodicity conditions on the faces at $Z = \pm a_1$ is more difficult. Theses faces cannot be let free because normal strains will develop and the applied boundary condition $U3 = 2a_3$ would not lead to a unit average strain. Although the average deformation at the mesoscale (composite) is pure shear γ_{13}, the deformation at the microscale (fiber and matrix) would include components $E11, E22$, and $E33$. Therefore, periodic conditions on the faces at $Z = \pm a_1$ must be enforced. Since there is no applied strain $E33$, we can use tie constraints.

Tie constraints are applied in the interaction module, as follows.

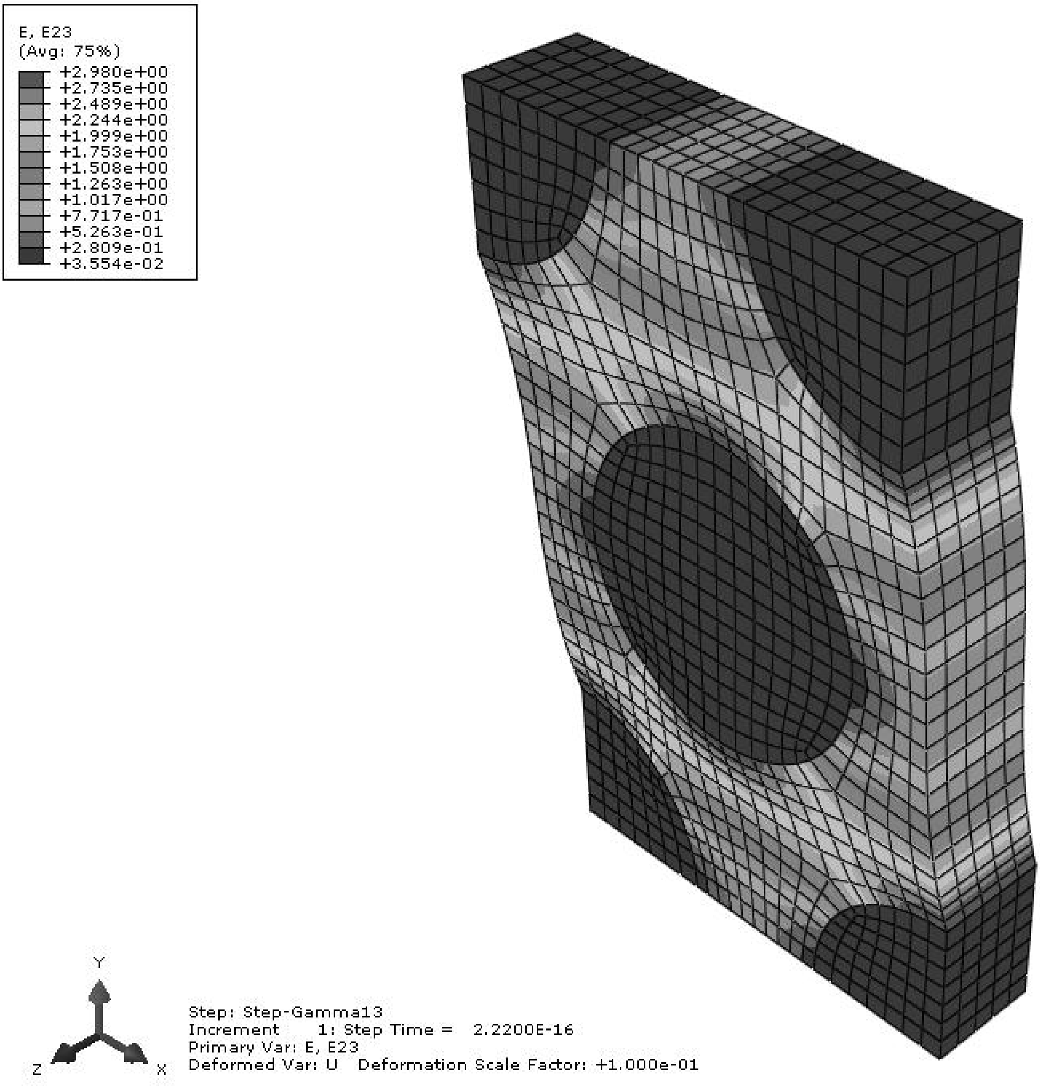


Fig. 6.7: Deformed shape and contour plot of strain E23 for Example 6.3. Note the coordinate system X,Y,Z.

Table 6.3: In-plane shear modulus of the unidirectional lamina.

Property	PMM	FEA
G_{12} $[MPa]$	2,594	2,579

```
Module: Interaction
Menu:   Constraint, Create
    Type: Tie, Cont
    Choose the master type: Surface
    # pick: front face (surfaces z=2*a_1), Done
    Choose the slave type: Surface
    # pick: back face (surfaces z=0), Done
    Position Tolerance: Specify distance: [2.635] # 2*a_1
    # Uncheckmark: Adjust slave surface initial position
    # Uncheckmark: Tie rotational DOF if applicable
```

It is very important to specify the distance between the surfaces (position tolerance) so that CAE can find the nodes on both surfaces.

You can extract the Python script corresponding to the pseudo code shown above by looking at the `.rec` *file. The complete script to solve this example is available in [5, Ex_6.3-TC.py]. The script can be executed inside* Abaqus*/CAE by doing* `Menu: File, Run script: Ex_6.3-TC.py`*. It includes [5, srecover.py] to calculate the average strains and average stresses, from which the result displayed in Table 6.3 is obtained.*

6.3 Local-Global Analysis

In local-global analysis (Figure 6.8), an RVE is used to compute the stress for a given strain at each Gauss integration point in the *global model*. The global model is used to compute the displacements and resulting strains, assuming that the material is homogeneous. The *local model* takes the inhomogeneities into account by modeling them with an RVE.

Equations (6.13-6.15) are used in Section 6.2 to enforce one component of strain at a time, with the objective of finding the equivalent elastic properties of the material. Equations (6.13-6.15) are still valid for a general state of strain applied to the RVE but care must be taken with the specification of periodic boundary conditions at the edges and vertices, as discussed in page 233. Equations (6.13-6.15) are nine constraint equations that can be imposed between all the pairs of periodic points on the faces of the RVE except on the edges and vertices.

On the faces $x_1 = \pm a_1$, u_1 is used to impose ε^0_{11}, u_2 is used to impose $\varepsilon^0_{21} = \gamma_6/2$, and u_3 is used to impose $\varepsilon^0_{31} = \gamma_5/2$. To achieve this, (6.13) is expanded into its three components, using tensor notation for strains, as follows

$$\begin{aligned}
u_1(a_1,x_2,x_3) - u_1(-a_1,x_2,x_3) &= 2a_1\varepsilon^0_{11} \\
u_2(a_1,x_2,x_3) - u_2(-a_1,x_2,x_3) &= 2a_1\varepsilon^0_{21} \\
u_3(a_1,x_2,x_3) - u_3(-a_1,x_2,x_3) &= 2a_1\varepsilon^0_{31}
\end{aligned} \tag{6.42}$$

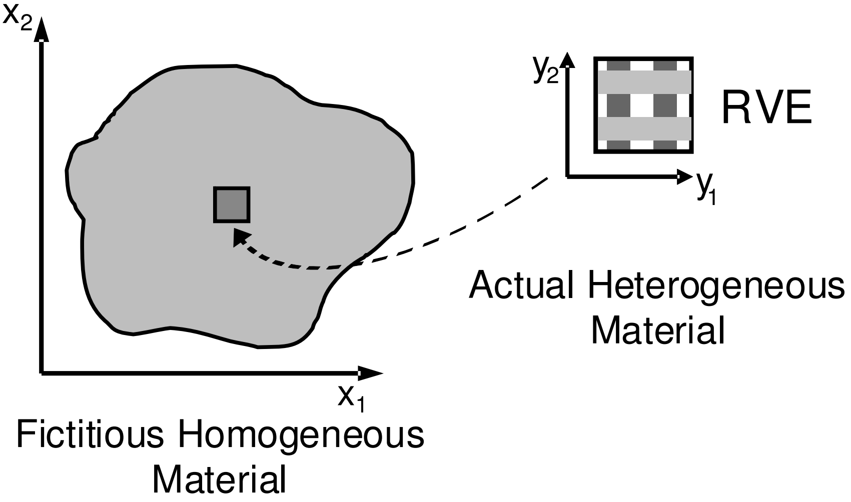

x2
x1
y2
y1
RVE
Actual Heterogeneous Material
Fictitious Homogeneous Material

Fig. 6.8: Local-global analysis using RVE.

On the faces $x_2 = \pm a_2$, u_1 is used to impose $\varepsilon^0_{12} = \gamma_6/2$, u_2 is used to impose ε^0_{22}, and u_3 is used to impose $\varepsilon^0_{32} = \gamma_4/2$. Therefore, (6.14) is expanded into its three components, using tensor notation for strains, as follows

$$\begin{aligned}
u_1(x_1, a_2, x_3) - u_1(x_1, -a_2, x_3) &= 2a_2\varepsilon_{12} \\
u_2(x_1, a_2, x_3) - u_2(x_1, -a_2, x_3) &= 2a_2\varepsilon_{22} \\
u_3(x_1, a_2, x_3) - u_3(x_1, -a_2, x_3) &= 2a_2\varepsilon_{32}
\end{aligned} \tag{6.43}$$

On the faces $x_3 = \pm a_3$, u_1 is used to impose $\varepsilon^0_{13} = \gamma_5/2$, u_2 is used to impose $\varepsilon^0_{23} = \gamma_4/2$, and u_3 is used to impose ε^0_{33}. Therefore, (6.15) is expanded into its three components, using tensor notation for strains, as follows

$$\begin{aligned}
u_1(x_1, x_2, a_3) - u_1(x_1, x_2, -a_3) &= 2a_3\varepsilon_{13} \\
u_2(x_1, x_2, a_3) - u_2(x_1, x_2, -a_3) &= 2a_3\varepsilon_{23} \\
u_3(x_1, x_2, a_3) - u_3(x_1, x_2, -a_3) &= 2a_3\varepsilon_{33}
\end{aligned} \tag{6.44}$$

Since each edge belongs to two faces, on every edge, it would seem that each component of displacement would be used to impose two CE, one from each face, as given by (6.42-6.44). However, as discussed in page 233, only one CE can be written for each component of displacement. Therefore, edges must be dealt with separately. Similarly, since three faces converge at a vertex, three periodic CE, one from each face, need to be imposed using a single component of displacement. Following a derivation similar to that presented in page 233, the following is obtained.

The planes $x_1 = \pm a_1$ and $x_2 = \pm a_2$ define two pairs of edges, for which (6.42-6.44) reduce to the following six equations (with $i = 1, 2, 3$), as follows

$$\begin{aligned}
u_i(+a_1,+a_2,x_3)-u_i(-a_1,-a_2,x_3)-2a_1\varepsilon_{i1}-2a_2\varepsilon_{i2} &= 0\\
u_i(+a_1,-a_2,x_3)-u_i(-a_1,+a_2,x_3)-2a_1\varepsilon_{i1}+2a_2\varepsilon_{i2} &= 0
\end{aligned}\tag{6.45}$$

The planes $x_1 = \pm a_1$ and $x_3 = \pm a_3$ define two pairs of edges, for which (6.42-6.44) reduce to the following six equations (with $i = 1, 2, 3$), as follows

$$\begin{aligned}
u_i(+a_1,x_2,+a_3)-u_i(-a_1,x_2,-a_3)-2a_1\varepsilon_{i1}-2a_3\varepsilon_{i3} &= 0\\
u_i(+a_1,x_2,-a_3)-u_i(-a_1,x_2,+a_3)-2a_1\varepsilon_{i1}+2a_3\varepsilon_{i3} &= 0
\end{aligned}\tag{6.46}$$

The planes $x_2 = \pm a_2$ and $x_3 = \pm a_3$ define two pairs of edges, for which (6.42-6.44) reduce to the following six equations (with $i = 1, 2, 3$), as follows

$$\begin{aligned}
u_i(x_1,+a_2,+a_3)-u_i(x_1,-a_2,-a_3)-2a_2\varepsilon_{i2}-2a_3\varepsilon_{i3} &= 0\\
u_i(x_1,+a_2,-a_3)-u_i(x_1,-a_2,+a_3)-2a_2\varepsilon_{i2}+2a_3\varepsilon_{i3} &= 0
\end{aligned}\tag{6.47}$$

Four pairs of corners need to be analyzed one at a time. For each pair, the corners are located symmetrically with respect to the center of the RVE located at coordinates $(0, 0, 0)$. The resulting CE are as follows

$$\begin{aligned}
u_i(+a_1,+a_2,+a_3)-u_i(-a_1,-a_2,-a_3)-2a_1\varepsilon_{i2}-2a_2\varepsilon_{i2}-2a_3\varepsilon_{i3} &= 0\\
u_i(+a_1,+a_2,-a_3)-u_i(-a_1,-a_2,+a_3)-2a_1\varepsilon_{i2}-2a_2\varepsilon_{i2}+2a_3\varepsilon_{i3} &= 0\\
u_i(-a_1,+a_2,+a_3)-u_i(+a_1,-a_2,-a_3)+2a_1\varepsilon_{i2}-2a_2\varepsilon_{i2}-2a_3\varepsilon_{i3} &= 0\\
u_i(+a_1,-a_2,+a_3)-u_i(-a_1,+a_2,-a_3)-2a_1\varepsilon_{i2}+2a_2\varepsilon_{i2}-2a_3\varepsilon_{i3} &= 0
\end{aligned}\tag{6.48}$$

Example 6.4 *Apply $\epsilon_2^0 = 0.2\%$ and $\gamma_4^0 = 0.1\%$ simultaneously to the composite in Example 6.2 (p. 226).Compute the average $\overline{\sigma}_2$ and $\overline{\sigma}_{12}$ in the RVE and the maximum stress σ_2 and σ_{12} anywhere in the RVE.*

Solution to Example 6.4 *Tie constraints cannot be used as a substitute for periodic conditions when the strain to be applied between two faces is not zero. Symmetry boundary conditions cannot be used because the applied deformation $\epsilon_2^0 = 0.002$ and $\gamma_{23}^0 = 0.001$ is not symmetric with respect to the coordinate planes (x_1, x_2) and (x_1, x_3) $(Z, X$ and Z, Y in the model shown in Figure 6.9). Instead, the full RVE shown in Figure 6.4 must be used along with the CE explained in Eqs. (6.30-6.39). However, 2D model can be used because the geometry is prismatic, i.e., all cross sections normal to the x_1-axis are identical and deform identically. The periodic conditions on the faces at $\pm a_1$ with $\epsilon_1 = \gamma_{12} = \gamma_{13} = 0$ are represented on a 2D model by a plane strain condition, i.e., using plane strain elements.*

The fiber diameter is $d_f = 7\ \mu m$ and the RVE dimensions are:

$$a_1 = a_2/4 = 1.3175\ \mu m \quad ; \quad a_2 = 5.270\ \mu m \quad ; \quad a_3 = 9.128\ \mu m$$

The geometry (`Module: Part`), material and sections (`Module: Property`), assembly (`Module: Assembly`), step (`Module: Step`), and mesh (`Module: Mesh`) can be created

Table 6.4: Average stress in the RVE and Maximum stress on the matrix for Example 6.4.

Stress Component	Model Labels	Direction	Average	Maximum in Matrix
σ_{11}	S33	Fiber	9.920	16.710
σ_{22}	S11	Transverse	21.534	28.710
σ_{33}	S22	Transverse	11.841	16.230
σ_{23}	S12	Transverse	2.425	5.640

in the same fashion as it was done in Example 6.2 but for a 2D model representing the RVE with one full fiber and four quarter fibers as shown in Figure 6.4. No symmetry boundary conditions can be used in this case. Instead, periodic boundary conditions (PBC) (6.13-6.15) must be enforced in `Module: Load`*. These PBC are enforced by using constraint equations (CE). There are no loads per se, but rather the constant terms on the right-hand side (RHS) of (6.13-6.15), which are generated by the applied strains $\epsilon_2^0, \epsilon_3^0, \epsilon_{23}^0$, yield the loading for the model.*

To write the CEs, one needs to define a reference point, as well as master and slave nodes. Then, the master and slave nodes are connected by constraint equations representing exactly (6.13-6.15). As noted earlier, the equations for the vertices are dealt with separately from the equations for the faces. The task is automated in [5, PBC_2D.py].

The complete script to solve this example is available in [5, Ex_6.4-CE.py]. First, it generates the mesh and sets of nodes later to be used in the CE. The script can be executed inside Abaqus/*CAE by doing* `Menu: File, Run script, [Ex_6.4-CE.py], OK`*. It calls [5, PBC_2D.py] to write the constraint equations and [5, srecover2D.py] to calculate the average stresses. The results are shown in Table 6.4.*

6.4 Laminated RVE

A similar procedure to that used to obtain the RVE at the microscale can be used to analyze laminates on the mesoscale. In this case the RVE represents a laminate. Therefore, the through-thickness direction should remain free to expand along the thickness. For example, with laminas parallel to the x-y plane, then $\sigma_z = 0$ and (6.15) is not enforced, so that the thickness coordinate is free to contract (see Figure 6.10). In general, the RVE must include the whole thickness. For symmetrical laminates subjected to in-plane loads, the RVE can be defined with half the thickness using symmetry boundary conditions (see Example 6.5, p. 243).

The CE for a laminated RVE are simpler. Only (6.13) and (6.14) must be enforced. In an hexahedral RVE, such as shown in Figure 6.10, only four faces ($x_1 = \pm a_1$ and $x_2 = \pm a_2$) and the four edges defined by these faces need to be considered.

Therefore, in a laminated RVE the constraint equations (6.13) and (6.14) become the following. On the periodic pair of faces $x_1 = \pm a_1$, the CE are derived from (6.13) as

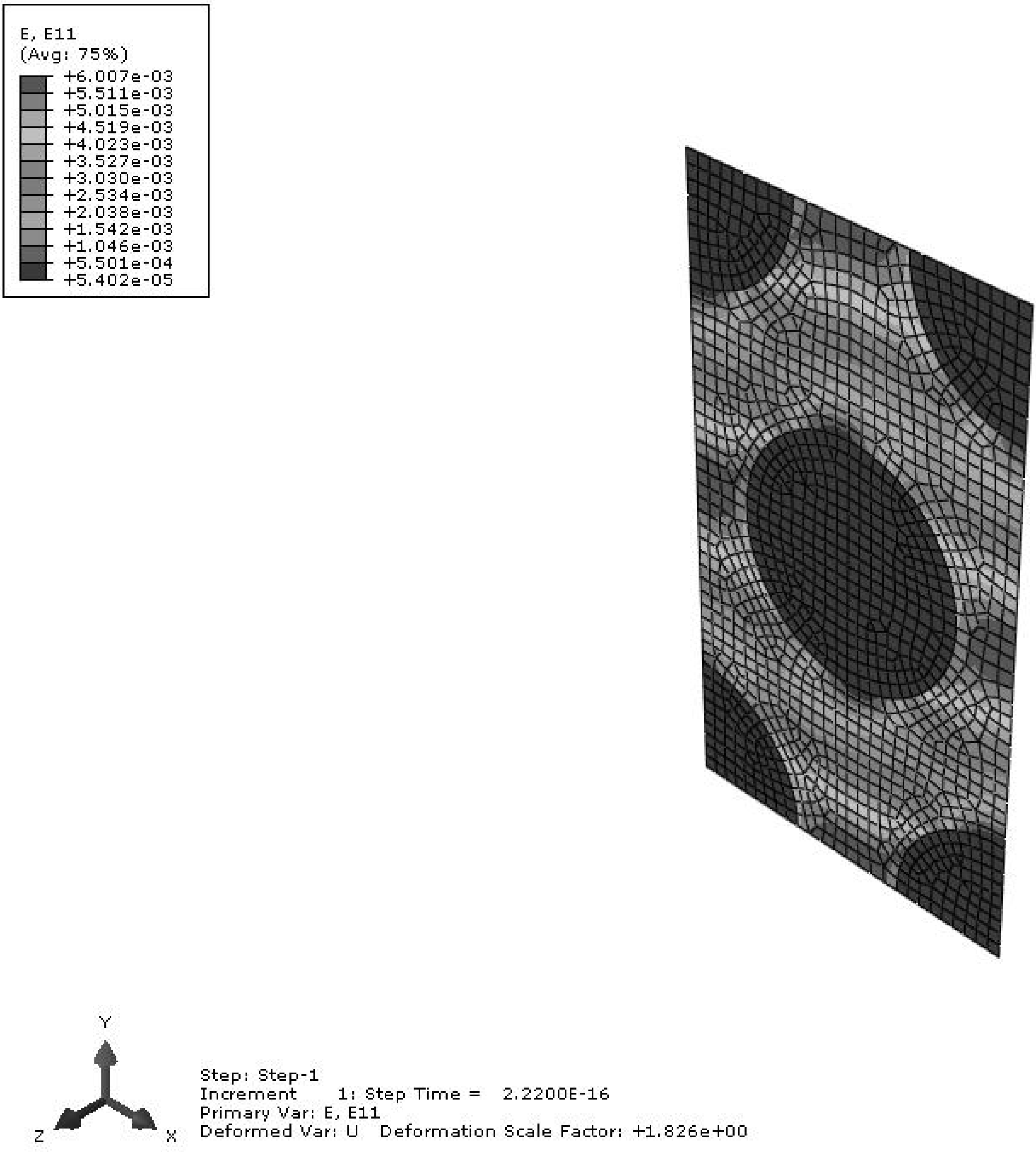


Fig. 6.9: Contour plot of ϵ_{22} (E11 in model coordinates X, Y, Z) for Example 6.4.

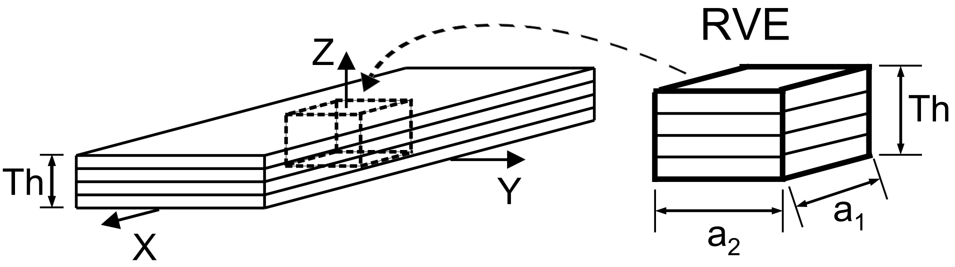


Fig. 6.10: Laminated RVE.

$$\begin{aligned}
u_1(a_1,x_2,x_3) - u_1(-a_1,x_2,x_3) - 2a_1\varepsilon_{11} &= 0 \\
u_2(a_1,x_2,x_3) - u_2(-a_1,x_2,x_3) - 2a_1\varepsilon_{21} &= 0 \\
u_3(a_1,x_2,x_3) - u_3(-a_1,x_2,x_3) - 2a_1\varepsilon_{31} &= 0
\end{aligned} \tag{6.49}$$

and on the pair of faces $x_2 = \pm a_2$, the CE are derived from (6.14)

$$\begin{aligned}
u_1(x_1,a_2,x_3) - u_1(x_1,-a_2,x_3) - 2a_2\varepsilon_{12} &= 0 \\
u_2(x_1,a_2,x_3) - u_2(x_1,-a_2,x_3) - 2a_2\varepsilon_{22} &= 0 \\
u_3(x_1,a_2,x_3) - u_3(x_1,-a_2,x_3) - 2a_2\varepsilon_{32} &= 0
\end{aligned} \tag{6.50}$$

The planes $x_1 = \pm a_1$ and $x_2 = \pm a_2$ define two pairs of periodic edges, for which (6.13-6.14) reduce to the following equations

$$\begin{aligned}
u_1(+a_1,+a_2,x_3) - u_1(-a_1,-a_2,x_3) - 2a_1\varepsilon_{11} - 2a_2\varepsilon_{12} &= 0 \\
u_2(+a_1,+a_2,x_3) - u_2(-a_1,-a_2,x_3) - 2a_1\varepsilon_{21} - 2a_2\varepsilon_{22} &= 0 \\
u_3(+a_1,+a_2,x_3) - u_3(-a_1,-a_2,x_3) - 2a_3\varepsilon_{31} &= 0
\end{aligned} \tag{6.51}$$

and

$$\begin{aligned}
u_1(+a_1,-a_2,x_3) - u_1(-a_1,+a_2,x_3) - 2a_1\varepsilon_{11} + 2a_2\varepsilon_{12} &= 0 \\
u_2(+a_1,-a_2,x_3) - u_2(-a_1,+a_2,x_3) - 2a_1\varepsilon_{21} + 2a_2\varepsilon_{22} &= 0 \\
u_3(+a_1,-a_2,x_3) - u_3(-a_1,+a_2,x_3) - 2a_3\varepsilon_{32} &= 0
\end{aligned} \tag{6.52}$$

For in-plane analysis, $\varepsilon_{31} = \varepsilon_{32} = 0$ and the third equation in (6.49)-(6.52) are automatically satisfied.

Example 6.5 *Compute G_{xy} for a $[0/90/-45/45]_S$ laminate with properties $E_1 = 139$ GPa, $E_2 = 14.5$ GPa, $G_{12} = G_{13} = 5.86$ GPa, $G_{23} = 2.93$ GPa, $\nu_{12} = \nu_{13} = 0.21$, $\nu_{23} = 0.38$, and lamina thickness $t_k = 1.25mm$.*

Solution to Example 6.5 *As a result of laminate symmetry, a representative volume element (RVE) of one half the laminate thickness and symmetry boundary conditions at $z = 0$ can be used. The model of the RVE is discretized with 3D solid elements. The pseudo code below can be followed to build the model using the* Abaqus *CAE GUI.*

i. *Setting the* `Work Directory`

```
Menu:   File, Set Work Directory,  [C:\Simulia\User\Ex_6.5], OK
Menu:   File, Save As, [C:\Simulia\User\Ex_6.5\Ex_6.5.cae], OK
```

ii. *Creating parts*

```
Module: Part
Menu:   Part, Create
    3D, Deformable, Solid, Extrusion, Approx. size [10], Cont
Menu:   Add, Line, Rectangle, [0,0], [1,1], X, Done, Depth [5], OK
Menu:   Tools, Datum
    Type: Plane, Method: Offset from principal plane
    XY Plane, Offset [1.25]
    XY Plane, Offset [2.5]
    XY Plane, Offset [3.75], X, # close Create Datum pop-up window
Menu:   Tools, Partition
    Cell, Use datum plane, # pick: Datum plane z=1.25,
    Create Partition
    # pick: larger cell, Done, # pick: Datum plane z=2.5,
    Create Partition
    # pick: larger cell, Done, # pick: Datum plane z=3.75,
    Create Partition
    Done, # close Create partition pop-up window
```

iii. *Defining materials, sections, and assigning sections to parts*

```
Module: Property
Menu:   Material, Create
    Mechanical, Elasticity, Elastic, Type: Engineering Constants
    [139000, 14500, 14500, 0.21, 0.21, 0.38, 5860, 5860, 2930], OK
Menu:   Assign, Material Orientation
    # pick: layer 1 on Z=4*1.25 (free surface), Done
    Use Default Orientation or Other Method
    Definition: Coordinate system,
    Additional Rotation Direction: Axis 3
    Additional Rotation: Angle [0], OK
    # pick: layer 2 on Z=3*1.25, Done
    Use Default Orientation or Other Method
    Definition: Coordinate system,
    Additional Rotation Direction: Axis 3
    Additional Rotation: Angle [90], OK
    # pick: layer 3 on Z=2*1.25, Done
    Use Default Orientation or Other Method
    Definition: Coordinate system,
    Additional Rotation Direction: Axis 3
    Additional Rotation: Angle [-45], OK
    # pick: layer 4 on Z=1*1.25, Done
    Use Default Orientation or Other Method
    Definition: Coordinate system,
    Additional Rotation Direction: Axis 3
    Additional Rotation: Angle [45], OK, Done
Menu:   Section, Create
    Solid, Homogeneous, Cont, Material: Material-1, OK
Menu:   Assign, Section, # select all, Done, OK, Done
```

iv. *Creating assembly*

```
Module: Assembly
Menu:   Instance, Create, Independent, OK
```

v. *Defining analysis steps*

```
Module: Step
Menu:   Step, Create
    Procedure type: Linear perturbation, Static, Cont, OK
Menu:   Output, Field Output Requests, Edit, F-Output-1
```

```
        Output Variables: [S,E,U,IVOL], OK
```

vi. *Adding BC and loads*
An in-plane shear strain $\gamma_{xy}^0 = 1$ must be applied to the RVE, and the model solved. From the results, the laminate shear stiffness G_{xy} can be obtained directly by computing the average stress in the RVE. The boundary conditions (6.49-6.52) are introduced as follows:

```
# Apply simple shear
Module: Load
Menu:   BC, Manager
    Create, Name [yfix], Step: Initial, Disp/Rota, Cont
    # pick: surface @ y=0, Done, # Checkmark: U1 and U2, OK
    Create, Name [disp], Step: Step-1, Disp/Rota, Cont
    # pick: surface @ Y=1, Done
    # checkmark: U1 [1], # checkmark: U2 [0], OK
    Create, Name [zsymm], Step: Initial, Symm/Anti/Enca, Cont
    # pick: surfaces @ Z=0, Done, ZSYMM, OK
    # close BC Manager pop-up window
```

vii. *Defining constraints*
Apply Tie constraint x = constant surfaces to get $\epsilon_{xx} = 0$

```
Module: Interaction
Menu:   Constraint, Create, Tie, Cont
    Choose the master type: Surface, # pick: surface x=0, Done
    Choose the slave type: Surface, # pick: surface x=1, Done
    Position Tolerance: Specify distance [1]
    # Uncheckmark: Adjust slave surface initial position # !!!
    # Uncheckmark: Tie rotational DOFs if applicable, OK
```

viii. *Meshing the model*

```
Module: Mesh
Menu:   Seed, Instance
    Approx global size [1.25], OK
    # one element/layer-thickness enough for pure shear
Menu:   Mesh, Instance, Yes
```

ix. *Solving and visualizing the results*

```
Module: Job
Menu:   Job, Manager
    Create, Cont, OK, Submit, # when Completed, Results
Menu:   File, Save
Menu:   File, Run Script, [Ex_6.5-post.py], OK
```

The script [5, Ex_6.5-post.py] can be used to calculate the average shear stress, and thus the shear modulus. On account of the applied strain being equal to unity, the computed average stress σ_{12} is equal to C_{66}. Therefore, the last line in the script yields $G_{12} = 21,441\ MPa$.

```
""" Post-processing Ex. 6.5 """

# make sure the Work Directory is OK
import os
os.chdir(r'C:\Simulia\User\Chapter_6\Ex_6.5')

# Open the Output Database for the current Job
from visualization import *
odb = openOdb(path='Job-1.odb');
myAssembly = odb.rootAssembly;

# Create temporary variable to hold frame repository speeds up the process
frameRepository = odb.steps['Step-1'].frames;
frameS=[];
frameIVOL=[];

# Create a Coordinate System in the Laminate direction (Global)
coordSys = odb.rootAssembly.DatumCsysByThreePoints(name='CSYSLAMINATE',
    coordSysType=CARTESIAN, origin=(0,0,0),
    point1=(1.0, 0.0, 0), point2=(0.0, 1.0, 0.0) )

# Transform stresses from Lamina Coordinates
# to Laminate Coordinate System defined in CSYSLAMINATE
# stressTrans=odb.steps['Step-1'].frames[-1].fieldOutputs['S']\
#   .getTransformedField(datumCsys=coordSys)
stressTrans=frameRepository[-1].fieldOutputs['S']\
    .getTransformedField(datumCsys=coordSys)

# Insert transformed stresses into frameS
frameS.insert(0,stressTrans.getSubset(position=INTEGRATION_POINT));
frameIVOL.insert(0,frameRepository[-1].fieldOutputs['IVOL']\
    .getSubset(position=INTEGRATION_POINT));

Tot_Vol=0;      # Total Volume
Tot_Stress=0;   # Stress Sum

for II in range(0,len(frameS[-1].values)):
     Tot_Vol=Tot_Vol+frameIVOL[0].values[II].data;
     Tot_Stress=Tot_Stress+frameS[0].values[II].data * frameIVOL[0]\
        .values[II].data;

#Calculate Average
Avg_Stress=Tot_Stress/Tot_Vol;
print 'Abaqus/Standard Stress Tensor Order:'
# From Abaqus Analysis User's Manual - 1.2.2 Conventions
print ' 11-22-33-12-13-23';
print Avg_Stress;

print 'G12=',Avg_Stress[3]
```

Suggested Problems

Problem 6.1 *Consider a unidirectional composite with isotropic fibers* $E_f = 241$ *GPa*, $\nu_f = 0.2$, *and isotropic matrix* $E_m = 3.12$ *GPa*, $\nu_m = 0.38$ *with fiber volume fraction* $V_f = 0.4$. *The fiber diameter is* $d_f = 7$ μm, *placed in a square array as shown in Figure 6.1. Choose an RVE including one full fiber in the center, with vertical faces spaced* $2a_2$ *and horizontal faces spaced* $2a_3$. *Part a: compute the first 3 columns of the stiffness matrix in (6.10). Part b: compute* C_{66}.

Problem 6.2 *Consider the same material and fiber distribution of Problem 6.1, but choose an RVE with faces rotated* 45° *with respect to the horizontal and vertical direction in Figure 6.1. Therefore, the RVE size will be* $2\sqrt{2}a_2$ *and horizontal faces* $2\sqrt{2}a_3$ *and it will include two fibers (one full and four quarters). Be careful to select a correct RVE that is periodic.*

Part a: compute the first 3 columns of the stiffness matrix in (6.10).

Part b: compute C_{66}.

Problem 6.3 *Compute* $E_1, E_2, \nu_{12}, \nu_{23}, G_{12}$, *and* G_{23} *using the stiffness matrices calculated in Problem 6.1.*

Problem 6.4 *Compute* $E_1, E_2, \nu_{12}, \nu_{23}, G_{12}$, *and* G_{23} *using the stiffness matrices calculated in Problem 6.2.*

Problem 6.5 *Show that removing the tie constraints from Example 6.3 (p. 236) leads to nonzero normal strains and to a wrong result for* G_{13}.

Problem 6.6 *Compute* G_{12} *as in Example 6.3 (p. 236) but using symmetry boundary conditions to discretize only one quarter of the RVE.*

Problem 6.7 *Compute* G_{23} *by modifying the boundary conditions and tie constraints of Example 6.3, p. 236.*

Problem 6.8 *Compute* G_{23} *by using Example 6.4 (p. 240) as a guide.*

Chapter 7

Viscoelasticity

Our interest in viscoelasticity is motivated by observed creep behavior of polymer matrix composites (PMC), which is a manifestation of viscoelasticity. The time dependent response of materials can be classified as elastic, viscous, and viscoelastic. On application of a sudden load, which is then held constant, an elastic material undergoes instantaneous deformation. In a *one-dimensional state of stress*, the elastic strain is $\varepsilon = D\sigma$, where $D = 1/E$ is the compliance or inverse of the modulus E. The deformation then remains constant. Upon unloading, the elastic strain reverses to its original value; thus all elastic deformation is recovered.

The viscous material flows at a constant rate $\dot{\varepsilon} = \sigma/\eta$ where $\eta = \tau E_0$ is the Newton viscosity, E_0 is the initial modulus, and τ is the time constant of the material. The accumulated strain $\varepsilon = \int \dot{\varepsilon} dt$ cannot be recovered by unloading.

A viscoelastic material combines the behavior of the elastic and viscous material in one, but the response is more complex than just adding the viscous strain to the elastic strain. Let H be the Heaviside function defined as

$$\begin{aligned} H(t-t_0) &= 0 \text{ when } t < t_0 \\ H(t-t_0) &= 1 \text{ when } t \geq t_0 \end{aligned} \tag{7.1}$$

Upon step loading $\sigma = H(t-t_0)\ \sigma_0$, with a constant load σ_0, the viscoelastic material experiences a sudden elastic deformation, just like the elastic material. After that, the deformation grows by a combination of recoverable and unrecoverable viscous flow.

A simple series addition of viscous flow and elastic strain (*Maxwell model*, Figure 7.1(a), with $\eta = \tau E_0$) yields totally unrecoverable viscous flow plus recoverable elastic deformation as

$$\dot{\varepsilon}(t) = \frac{\sigma(t)}{\tau E_0} + \frac{\dot{\sigma}(t)}{E_0} \tag{7.2}$$

A simple parallel combination of elastic and viscous flow (*Kelvin model*, Figure 7.1(b), with $\eta = \tau E$) yields totally recoverable deformation with no viscous flow as

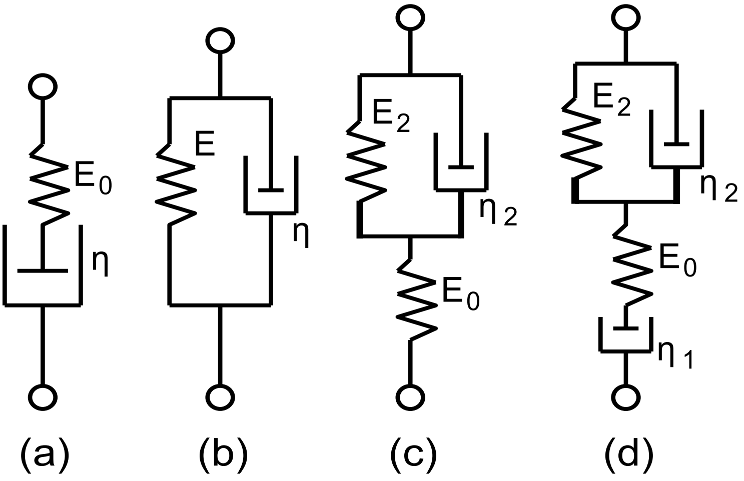


Fig. 7.1: Viscoelastic modes: (a) Maxwell, (b) Kelvin, (c) standard solid, (d) Maxwell-Kelvin.

$$\sigma(t) = \tau E\dot{\varepsilon}(t) + E\varepsilon(t) \tag{7.3}$$

but the deformation does not recover instantaneously.

Materials with unrecoverable viscous flow, such as (7.2), are called *liquids* even though the flow may occur very slowly. Glass is a liquid material over the time span of centuries; the thickness of window panes in medieval cathedrals is thicker at the bottom and thinner at the top, thus revealing the flow that took place over the centuries under the load imposed by gravity. Materials with fully recoverable viscous deformations, such as (7.3), are called *solids.* We shall see that structural design is much easier with solid materials than with liquid materials.

Please take heed of the common misconception introduced in early mechanics of materials courses that most structural materials are elastic. Only perfectly crystalline materials are elastic. Most materials are viscoelastic if observed for sufficiently long periods of time, or at sufficiently high temperature. In other words, most real materials are viscoelastic.

For elastic materials, the compliance D is the inverse of the modulus E, both of which are constants, and they are related by

$$DE = 1 \tag{7.4}$$

For viscoelastic materials in the time domain, the compliance is called $D(t)$ and it is related to the time varying relaxation $E(t)$ in a similar but not so simple way, as it will be shown in Section 7.3. Note that the relaxation $E(t)$ takes the place of the modulus E. A brief derivation of the relationship between compliance and relaxation is presented next, in order to facilitate the presentation of viscoelastic models in Section 7.1. When both the compliance D and the relaxation E are functions of time, (7.4) simply becomes

$$D(t)E(t) = 1 \tag{7.5}$$

Both $D(t)$ and $E(t)$ are functions of time and thus it is not possible to operate algebraically on (7.5) to get either function explicitly in terms of the other. To find one from the other, take the Laplace transform (see Section 7.3) to get

$$s^2 D(s) E(s) = 1 \tag{7.6}$$

Since both $D(s)$ and $E(s)$ are algebraic functions of s, and the time t is not involved, it is possible to operate algebraically to get

$$E(s) = \frac{1}{s^2 D(s)} \tag{7.7}$$

Finally, the relaxation in the time domain is the inverse Laplace of (7.7) or

$$E(t) = L^{-1}[E(s)] \tag{7.8}$$

Similarly, the compliance $D(t)$ can be obtained from the relaxation $E(t)$ as

$$D(t) = L^{-1}\left[\frac{1}{s^2 L[E(t)]}\right] \tag{7.9}$$

where $L[\,]$ indicates the Laplace transform and $L^{-1}[\,]$ indicates the inverse Laplace transform.

7.1 Viscoelastic Models

The viscoelastic material models presented in this section are convenient curve fits of experimental data. In the time domain, the usual experiments are the creep and relaxation tests. In the *creep test*, a constant stress σ_0 is applied and the ensuing strain is measured. The ratio of measured strain to applied stress is the compliance $D(t) = \varepsilon(t)/\sigma_0$. In the *relaxation test*, a constant strain ε_0 is applied and the stress needed to maintain that strain is measured. The ratio of measured stress to applied strain is the relaxation $E(t) = \sigma(t)/\varepsilon_0$.

7.1.1 Maxwell Model

To derive the compliance of the *Maxwell* model [45], a creep test is performed under constant stress σ_0 applied at the ends of the model shown in Figure 7.1(a). The rate of strain is given by (7.2). Integrating with respect to time we get

$$\varepsilon(t) = \frac{1}{\tau E_0}\int_0^t \sigma_0 dt + \frac{\sigma_0}{E_0} \tag{7.10}$$

where E_0 is the elastic constant of the spring, τ is the time constant of the material, and $\eta_0 = \tau E_0$ in Figure 7.1(a). The spring and dashpot are subject to the same load and to the same constant stress σ_0, so evaluating the integral yields

$$\varepsilon(t) = \frac{\sigma_0\, t}{\tau E_0} + \frac{\sigma_0}{E_0} \tag{7.11}$$

Table 7.1: Some common Laplace transforms.

Function	**f(t)=L^{-1}{f(s)}**	**f(s)=L{f(t)}**
constant	a	a/s
linear	at	a/s^2
derivative	df/dt	sf(s)-f(0)
exponential	$\exp(at)$	$1/(s-a)$
convolution integral	$\int_0^t f(t-\tau)g(\tau)d\tau$	$L\{f\}L\{g\}$

Then, the compliance is

$$D(t) = \frac{1}{E_0} + \frac{t}{\tau E_0} \tag{7.12}$$

To derive the relaxation of the Maxwell model, take the Laplace transform of (7.12) (using Table 7.1 or *MATLAB*) to get

$$D(s) = \frac{1}{sE_0} + \frac{1}{s^2\tau E_0} = \frac{s\tau + 1}{s^2\tau E_0} \tag{7.13}$$

At $t = 0$, the dashpot does not move, so E_0 is also the initial elastic modulus of the material. Now, the relaxation in the Laplace domain is

$$E(s) = \frac{1}{s^2 D(s)} = \frac{\tau E_0}{s\tau + 1} \tag{7.14}$$

and the relaxation in the time domain is obtained by taking the inverse Laplace transform (using Table 7.1 or *MATLAB*) to get

$$E(t) = E_0 \exp(-t/\tau) \tag{7.15}$$

Note that at $t = \tau$, the relaxation decays to 36.8% of its initial value, and thus τ is called the *time constant* of the material.

7.1.2 Kelvin Model

For the *Kelvin* model, only the creep test is possible, since a relaxation test would require an infinitely large stress to stretch the dashpot in Figure 7.1(b) to a constant value in no time. For a creep test, a constant stress $\sigma = \sigma_0$ is applied. Then, (7.3) is an ordinary differential equation (ODE) in $\varepsilon(t)$, which is satisfied by $\varepsilon(t) = (\sigma_0/E)\,[1 - exp(-t/\tau)]$. Therefore, the compliance $D(t) = \varepsilon(t)/\sigma_0$ is

$$D(t) = 1/E_0[1 - \exp(-t/\tau)] \tag{7.16}$$

Using (7.8), the relaxation function can be written with the aid of the Heaviside step function $H(t)$ and the Dirac delta function $\delta(t)$ as follows

$$E(t) = EH(t) + E\tau\delta(t) \tag{7.17}$$

where $\delta(t - t_0) = \infty\ if\ \ t = t_0$ and zero for any other time. The following *MATLAB* code yields (7.17):

```
syms s complex; syms Dt Et t E tau real;
Dt=expand((1-exp(-t/tau))/E)
Ds=laplace(Dt)
Es=1/Ds/s^2
Et=ilaplace(Es)
```

7.1.3 Standard Linear Solid

To have an initial compliance $1/E_0$, a spring is added to the Kelvin model (Figure 7.1.c). Then, the compliance is

$$D(t) = 1/E_0 + 1/E_2 \left[1 - \exp\left(\frac{-t}{\tau_2}\right)\right] \tag{7.18}$$

and

$$E(t) = E_\infty + (E_0 - E_\infty) \exp\left(\frac{-t\,(E_0 + E_2)}{\tau_2 E_2}\right) \tag{7.19}$$

where $E_\infty = (E_0^{-1} + E_2^{-1})^{-1}$ is the equilibrium modulus as time goes to infinity.

To obtain a better correlation, more spring-dashpot elements are added in series, as in

$$D(t) = D_0 + \sum_{j=1}^{n} D_j \left[1 - exp\,(-t/\tau_j)\right] \tag{7.20}$$

where τ_j are the retardation times [17]. When $n \to \infty$,

$$D(t) = \int_0^\infty \Delta(\tau)\left[1 - exp\,(-t/\tau)\,d\tau\right] \tag{7.21}$$

where $\Delta(\tau)$ is the compliance spectrum [17].

7.1.4 Maxwell-Kelvin Model

A crude approximation of a liquid material is the *Maxwell-Kelvin* model, also called the four-parameter model, described by Figure 7.1(d). Since the Maxwell and Kelvin elements are placed in series, the compliance is found by adding the compliances of the two individual modes, as

$$D\,(t) = \frac{1}{E_0} + \frac{t}{\tau_1 E_0} + \frac{1}{E_2}\left[1 - exp\,(-t/\tau 2)\right] \tag{7.22}$$

where E_0 is the elastic modulus, τ_1 takes the place of τ in (7.12), and E_2, τ_2 take the place of E, τ in (7.16). The relaxation modulus is given by [45, page 28]

$$
\begin{aligned}
E\left(t\right) &= \left(P_1^2 - 4P_2\right)^{-1/2}\left[\left(q_1 - \frac{q_2}{T_1}\right) exp\left(-t/T_1\right) - \left(q_1 - \frac{q_2}{T_2}\right) exp\left(-t/T_2\right)\right] \\
\eta_1 &= E_0\tau_1 \quad ; \quad \eta_2 = E_0\tau_2 \\
q_1 &= \eta_1 \quad ; \quad q_2 = \frac{\eta_1\,\eta_2}{E_2} \\
T_1 &= \frac{1}{2P_2}\left[P_1 + \sqrt{P_1^2 - 4P_2}\right] \quad ; \quad T_2 = \frac{1}{2P_2}\left[P_1 - \sqrt{P_1^2 - 4P_2}\right] \\
P_1 &= \frac{\eta_1}{E_0} + \frac{\eta_1}{E_2} + \frac{\eta_2}{E_2} \quad ; \quad P_2 = \frac{\eta_1\,\eta_2}{E_0 E_2}
\end{aligned}
\tag{7.23}
$$

Another way to determine if a material is a liquid or a solid is to look at its long-term deformation. If the deformation is unbounded, then it is a liquid. If the deformation eventually stops, then it is a solid.

7.1.5 Power Law

Another model, which is popular to represent relatively short-term deformation of polymers, is the *power law*

$$
E(t) = At^{-n} \tag{7.24}
$$

The parameters A and n are adjusted with experimental data. The power law is popular because it fits well the short-time behavior of polymers and because fitting the data is very easy; just take logarithm on both sides of (7.24) so that the equation becomes that of a line, then fit the parameters using linear regression. The compliance is obtained by using (7.9) as

$$
\begin{aligned}
D(t) &= D_0 + D_c(t) \\
D_c(t) &= [A\Gamma(1-n)\Gamma(1+n)]^{-1}t^n
\end{aligned}
\tag{7.25}
$$

where Γ is the Gamma function [46], $D_0 = 1/E_0$ is the elastic compliance, and the subscript $()_c$ indicates the creep component of the relaxation and compliance functions.

7.1.6 Prony Series

Although the short-term creep and relaxation of polymers can be described well by the power law, as the time range becomes longer, a more refined model becomes necessary. One such model is the *Prony series*, which consists of a number n of decaying exponentials

$$
E(t) = E_\infty + \sum_{i=1}^{n} E_i \exp(-t/\tau_i) \tag{7.26}
$$

where τ_i are the relaxation times, E_i are the relaxation moduli, and E_∞ is the equilibrium modulus, if any exists. For example, a Maxwell material is a "liquid" and thus $E_\infty = 0$. The larger the τ_i the slower the decay is. Note that at $t = 0$, $E_0 = E_\infty + \sum E_i$. Equation (7.26) can be rewritten as

$$E(t) = E_\infty + \sum_1^n m_i E_0 \exp(-t/\tau_i) \tag{7.27}$$

where $m_i = E_i/E_0$.

The Prony series can be written in terms of the shear modulus and bulk modulus (G, k, see Section 1.12.5) as follows

$$\begin{aligned} G(t) &= G_\infty + \sum_1^n g_i G_0 \exp(-t/\tau_i) \\ k(t) &= k_\infty + \sum_1^n k_i k_0 \exp(-t/\tau_i) \end{aligned} \tag{7.28}$$

where $g_i = G_i/G_0$ and $k_i = k_i/k_0$, and G_0, k_0 are the initial values of shear modulus and bulk modulus, respectively. Noting that at $t = 0$, $G_\infty = G_0(1 - \sum_1^n g_i)$, and $k_\infty = k_0(1 - \sum_1^n k_i)$, the Prony series can be rewritten as

$$\begin{aligned} G(t) &= G_0\left(1 - \sum_1^n g_i\right) + \sum_1^n g_i G_0 \exp(-t/\tau_i) \\ k(t) &= k_0\left(1 - \sum_1^n k_i\right) + \sum_1^n k_i k_0 \exp(-t/\tau_i) \end{aligned} \tag{7.29}$$

Data are input to *Abaqus*/CAE in `Edit Material` by specifying separately the elastic properties (E_0 and ν_0) and the viscous properties (g_i, k_i, τ_i) for the *i-th* element of the Prony series. Internally, *Abaqus* calculates G_0 and k_0 using the values entered for E_0 and ν and equations (1.74) and (1.76). For most polymers and composites it is usual to assume that the Poisson's ratio does not change with time, which according to (1.74) and (1.76) is achieved by setting $k_i = g_i$. Also, if ν is constant over time, $m_i = g_i$ in (7.27).

Note that the Young modulus entered in `Edit Material` represents E_0 if `Moduli time scale (for viscoelasticity)` is set to `Instantaneous`. Otherwise, if `Moduli time scale (for viscoelasticity)` is set to `Long term`, then the Young modulus represents E_∞. Of course, one can be calculated from the other by using the relationship $E_0 = E_\infty/(1 - \sum m_i)$.

For example, a Maxwell material (7.15) can be modeled by a one term Prony series by setting $g_i = k_i = 1$. However, *Abaqus* cannot calculate a material with $E_\infty = 0$, so the data must be such that $\sum g_i < 1$. Thus, to model a Maxwell material one has to use $g_1 = k_1 = 0.999$.

7.1.7 Standard Nonlinear Solid

While the Prony series can fit any material behavior if a large number of terms are used, other models are more efficient for fitting purposes, if harder to manipulate mathematically. For example the *Standard Nonlinear Solid* model

$$D(t) = D_0 + D_1'[1 - \exp(-t/\tau)^m] \tag{7.30}$$

can approximate well the long-term compliance in the α-region of polymer creep [47]. At room temperature, this is the region of interest to structural engineers since it spans the range of time from seconds to years. In contrast the β-region [47] is of interest to sound and vibration experts, among others, since it spans the sub-second range of times. In other words, for long-term modeling, all compliance occurring in the β-region can be lumped in the term D_0, with D_1' representing all the compliance that could ever be accumulated in the α-region. Equation (7.30) has four parameters. When the data span short times, it may be impossible to determine all four parameters because the material behavior cannot be distinguished from a 3-parameter power law (7.31). This can be easily understood if (7.30) is expanded in a power series, truncated after the first term as follows [47]

$$D(t) = D_0 + D_1'(t/\tau)^m[1 - (t/\tau)^m + ...] \approx D_0 + D_1 t^m \ ; \ D_1 = D_1'/\tau \tag{7.31}$$

For short times, all higher order powers of t can be neglected. What remains is a modified power law with only three parameters. Note that for short times, the parameter τ is combined with D_1' to form D_1. If the data cover a short time, the fitting algorithm will not be able to adjust both τ and D_1' in (7.31); virtually any combination of τ and D_1' will work. That means that short-term data must be modeled by a smaller number of parameters, in this case three.

7.1.8 Nonlinear Power Law

All models described so far represent linear viscoelastic materials. In the context of viscoelasticity, linear means that the parameters in the model are not a function of stress (see Section (7.2.1)). That means that the deformation at any fixed time can be made proportionally larger by increasing the stress. If any of the parameters are a function of stress, the material is nonlinear viscoelastic. For example a *nonlinear power law* takes the form

$$\dot{\varepsilon} = At^B\sigma^D \tag{7.32}$$

Take logarithm to both sides of (7.32) to get a linear equation in two variables

$$y = \bar{A} + BX_1 + DX_2 \ ; \ \bar{A} = \log(A), X_1 = \log(t), X_2 = \log(\sigma) \tag{7.33}$$

that can be fitted with a multiple linear regression algorithm in *MATLAB*.

Although most materials are not linearly viscoelastic, they can be approximated as linear viscoelastic if the range of stress at which the structure operates is narrow.

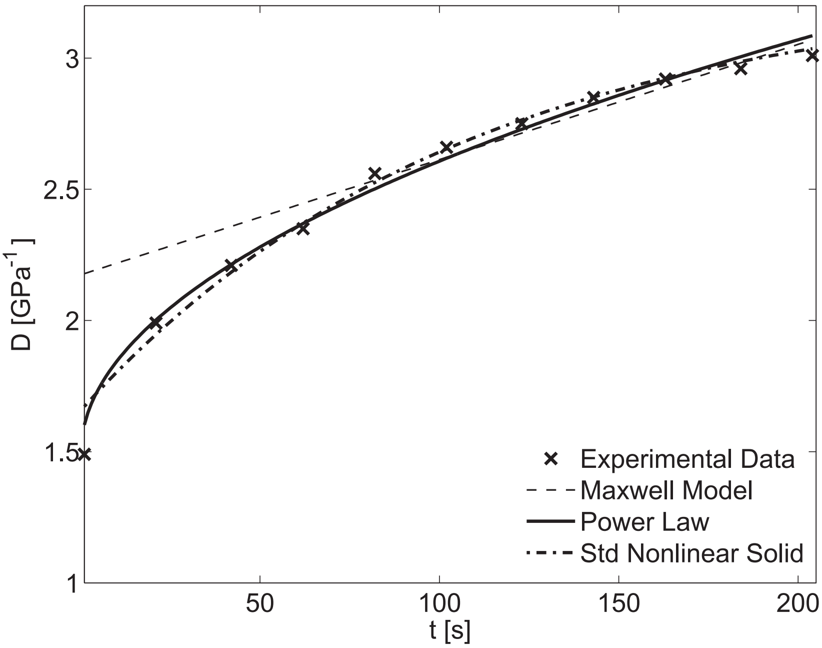


Fig. 7.2: Viscoelastic fit: Maxwell Model, Power Law, and Std Nonlinear Solid.

Example 7.1 *Fit the creep data in Table 7.2 with (a) Maxwell (7.12), (b) Power Law (7.31), and (c) Std Nonlinear Solid (7.30).*

Table 7.2: Creep data for Example 7.1.

time [sec]	1	21	42	62	82	102	123	143	163	184	204
D(t) [GPa^{-1}]	1.49	1.99	2.21	2.35	2.56	2.66	2.75	2.85	2.92	2.96	3.01

Solution to Example 7.1 *To fit the Maxwell model, fit a line to the secondary creep data; that is, ignore the curvy portion for short times to get $E_0 = 0.460$ GPa, $\tau = 495$ s.*

To fit the Power Law, write (7.19) as

$$D(t) - D_0 = D_1 t^m$$

where $D_0 = 1.49$ GPa^{-1} is the first datum in Table 7.2 (see also (7.31)). Take logarithm to both sides of above equation and adjust a line using linear regression to get $D_0 = 1.49$ GPa^{-1}, $D_1 = 0.1117$ (GPa sec)$^{-1}$, and $m = 0.5$.

To fit the Standard Nonlinear Solid you need to use a nonlinear solver to minimize the error between the predicted (expected) values e_i and the experimental (observed) values o_i. Such error is defined as the sum over all the available data points: $\chi^2 = \sum(e_i - o_i)^2/o_i^2$. In this way, the following are obtained: $D_0 = 1.657$ GPa^{-1}, $D_1' = 1.617$ GPa^{-1}, $\tau = 0.273$ sec, and $m = 0.0026$.

The experimental data and the fit functions are shown in Figure 7.2.

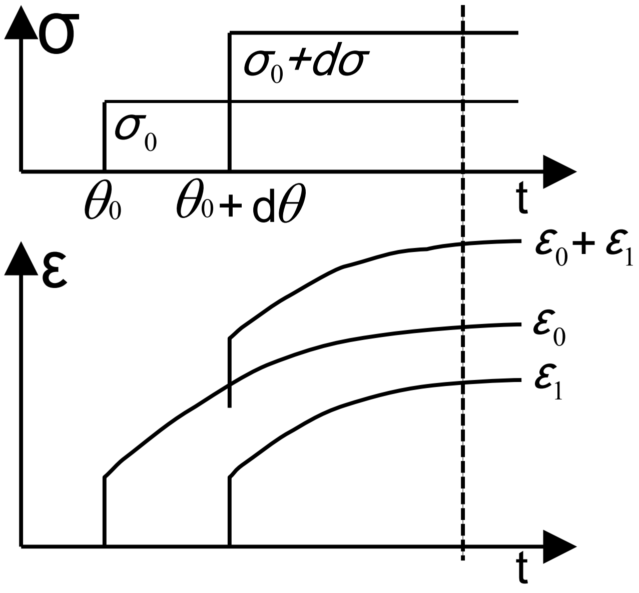


Fig. 7.3: Boltzmann superposition of strains.

7.2 Boltzmann Superposition

7.2.1 Linear Viscoelastic Material

A viscoelastic material is linear if superposition applies. That is, given a stress history

$$\sigma(t) = \sigma_1(t) + \sigma_2(t) \tag{7.34}$$

the strain is given by

$$\varepsilon(t) = \varepsilon_1(t) + \varepsilon_2(t) \tag{7.35}$$

where $\varepsilon_1(t)$, $\varepsilon_2(t)$ are the strain histories corresponding to $\sigma_1(t)$ and $\sigma_2(t)$, respectively. For linear materials, the creep compliance and relaxation modulus are independent of stress

$$\begin{aligned} D(t) &= \frac{\varepsilon(t)}{\sigma_0} \\ E(t) &= \frac{\sigma(t)}{\varepsilon_0} \end{aligned} \tag{7.36}$$

For nonlinear materials, $D(t,\sigma)$ is a function of stress and $E(t,\varepsilon)$ is a function of strain.

For a linear material subjected to a stress σ_0 applied at time $t=\theta_0$ (Figure 7.3) we have

$$\varepsilon(t) = \sigma_0 \; D(t,\theta_0) \quad ; \quad t > \theta_0 \tag{7.37}$$

Adding an infinitesimal load step $d\sigma$ at time $\theta_0 + d\theta$ results in

$$\varepsilon(t) = \sigma_0 \; D(t,\theta_0) + d\sigma \; D(t,\theta_0+d\theta) \quad ; \quad t > \theta_0 + d\theta \tag{7.38}$$

If stress changes continuously by $d\sigma$ over intervals $d\theta$, the summation (7.38) can be replaced by an integral to yield the accumulated strain as

$$\varepsilon(t) = \sigma_0 \; D(t, \theta_0) + \int_{\theta_0}^{t} D(t, \theta) d\sigma = \sigma_0 \; D(t, \theta_0) + \int_{\theta_0}^{t} D(t, \theta) \frac{d\sigma}{d\theta} d\theta \tag{7.39}$$

where the discrete times θ_0, $\theta_0 + d\theta$, etc., are represented by the continuous function θ. Although aging effects are negligible over each infinitesimal $d\theta$, they are significant over time. Therefore, the compliance $D(t, \theta)$ is a function of the current time t and all the time-history represented by θ in $D(t, \theta)$.

7.2.2 Unaging Viscoelastic Material

If the $\varepsilon_1(t, \theta)$ curve has the same shape as the $\varepsilon_1(t, \theta_0)$ curve, only translated horizontally, any curve can be shifted to the origin (Figure 7.3)

$$D(t, \theta) = D(t - \theta) \tag{7.40}$$

Equation (7.40) is the definition of unaging material. For a discussion of aging materials see [45, 48]. Equation (7.40) means that all the curves have the same shape regardless of age θ, only shifted. Note θ in (7.40) is a continuous function $\theta < t$ that denotes the time of application of each load (σ_0, $d\sigma$, etc.).

The response $\varepsilon(t, \theta)$ at a fixed time t is a function of the response at all times $\theta < t$. Therefore, it is said that the response is hereditary. If the material is aging, t and θ are independent variables in $D(t, \theta)$. For unaging materials, only one variable, $t - \theta$, is independent, so it does not matter how old the material is (t), it only matters for how long ($t - \theta$) it has been loaded with $d\sigma(\theta)$.

The creep compliance is the response of the material to stress and always starts when the stress is applied. If the change is gradual, from (7.39) we have

$$\varepsilon(t) = \int_{0}^{t} D(t - \theta) \; \dot{\sigma}(\theta) \; d\theta \tag{7.41}$$

The relaxation is

$$\sigma(t) = \int_{0}^{t} E(t - \theta) \; \dot{\varepsilon}(\theta) \; d\theta \tag{7.42}$$

The time dependent behavior of linear viscoelastic materials is hereditary, meaning that the behavior at time t depends on what happened to the material since the beginning of loading at $t = 0$.

Example 7.2 *Consider an unaging material represented by $D(t - \theta) = 1/E + (t - \theta)/\eta$ and loaded with (a) $\sigma_0 H(\theta)$ and (b) $\sigma_0 H(\theta - 1)$. Find $\varepsilon(t)$ in both cases and comment on the results.*

Solution to Example 7.2

$$\begin{aligned}
(\mathbf{a}) \quad & \sigma = \sigma_0 H(\theta) \;\Rightarrow\; d\sigma/dt = \sigma_0 \delta(0) \\
& \varepsilon(t) = \textstyle\int_0^t \left[\frac{1}{E} + \frac{(t-\theta)}{\eta} \right] \sigma_0 \delta(0) d\theta \\
& \varepsilon(t) = \left[\frac{1}{E} + \frac{t}{\eta} \right] \sigma_0 \; ; \; t > 0
\end{aligned}$$

$$\begin{aligned}
(\mathbf{b}) \quad & \sigma = \sigma_0 H(\theta - 1) \;\Rightarrow\; d\sigma/dt = \sigma_0 \delta(1) \\
& \varepsilon(t) = \textstyle\int_0^t \left[\frac{1}{E} + \frac{(t-\theta)}{\eta} \right] \sigma_0 \delta(1) d\theta \\
& \varepsilon(t) = \left[\frac{1}{E} + \frac{(t-1)}{\eta} \right] \sigma_0 ; \; t > 1
\end{aligned}$$

It can be seen that (b) is identical to (a), only shifted; meaning that there is no aging.

7.3 Correspondence Principle

The Laplace transform of a function $f(t)$ in the time domain (t-domain) maps to the Laplace domain (s-domain) as $f(s)$. The Laplace transform is defined as

$$L[f(t)] = f(s) = \int_0^\infty \exp(-st) f(t) dt \tag{7.43}$$

Most of the time, the Laplace transform can be obtained analytically, just using a table of transforms, such as Table 7.1. Taking the Laplace transform of (7.41, 7.42) yields

$$\varepsilon(s) = L[D(t)] \; L[\dot{\sigma}(t)] = sD(s)\sigma(s) \tag{7.44}$$

$$\sigma(s) = L[E(t)] \; L[\dot{\varepsilon}(t)] = sE(s)\varepsilon(s) \tag{7.45}$$

Multiplying (7.44) times (7.45) yields

$$s^2 \; D(s)E(s) = 1 \tag{7.46}$$

or

$$s \; D(s) = [s \; E(s)]^{-1} \tag{7.47}$$

where it can be seen that $sD(s)$ is the inverse of $sE(s)$. This is analogous to (7.4) for elastic materials.

The correspondence principle states that all the equations of elasticity, available for elastic materials, are valid for linearly viscoelastic materials in the Laplace domain. This principle is the basis, for example, of the determination of creep and relaxation of polymer matrix composites in terms of fiber and matrix properties using standard micromechanics methods, as shown in Section 7.6.

The inverse mapping from the Laplace domain to the time domain

$$f(t) = L^{-1}(f(s)) \tag{7.48}$$

is more difficult to compute. Decomposition in partial fractions [49] is a useful technique to break up $f(s)$ into simpler component functions for which the inverse Laplace can be found analytically. Another useful technique is the convolution theorem defined in Table 7.1. Also, the limiting value theorems

$$\begin{aligned} f(0) &= \lim_{s\to\infty}[sF(s)] \\ f(\infty) &= \lim_{s\to 0}[sF(s)] \end{aligned} \tag{7.49}$$

can be used to evaluate the initial and final response of a material in the time domain directly in the Laplace domain. Otherwise, the inverse Laplace can be found numerically using [50] or by the collocation method described in [8, Appendix D].

The Carson transform is defined as

$$\hat{f}(s) = sf(s) \tag{7.50}$$

In the Carson domain, the constitutive equations (7.41-7.42) become

$$\begin{aligned} \varepsilon(s) &= \hat{D}(s)\sigma(s) \\ \sigma(t) &= \hat{E}(s)\varepsilon(s) \end{aligned} \tag{7.51}$$

which are analogous, in the Carson domain, to the stress-strain equations of elastic materials in the time domain. Furthermore, the relationship between compliance and relaxation becomes

$$\hat{D}(s) = 1/\hat{E}(s) \tag{7.52}$$

7.4 Frequency Domain

The Fourier transform maps the time domain into the frequency domain. It is defined as

$$F[f(t)] = f(\omega) = \int_{-\infty}^{\infty} \exp(-i\omega t) f(t)\ dt \tag{7.53}$$

and its inverse

$$f(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \exp(i\omega t) f(w) d\omega \tag{7.54}$$

Applying the Fourier transform to (7.41-7.42) yields

$$\begin{aligned} \varepsilon(\omega) &= D(\omega)\dot{\sigma}(\omega) \\ \sigma(\omega) &= E(\omega)\dot{\varepsilon}(\omega) \end{aligned} \tag{7.55}$$

and

$$D(\omega) = -\frac{1}{\omega^2 E(\omega)} \tag{7.56}$$

where $D(\omega) = D' + iD''$ and $E(\omega) = E' + iE''$ are complex numbers. Here D', D'' are the storage and loss compliances, and E', E'' are the storage and loss moduli.

Using standard complex analysis we get

$$\begin{aligned} D' &= \frac{E'}{E'^2 + E''^2} \\ D'' &= \frac{E''}{E'^2 + E''^2} \end{aligned} \tag{7.57}$$

The frequency domain has a clear physical meaning. If a sinusoidal stress $\sigma(\omega, t) = \sigma_0 \exp(-i\omega t)$ is applied to a viscoelastic material, it responds with an out-of-phase sinusoidal strain $\varepsilon(\omega, t) = \varepsilon_0 \exp(-i\omega t + \phi)$. Furthermore, the complex compliance is $D(\omega) = \varepsilon(\omega, t)/\sigma(\omega, t)$ and the complex relaxation is simply the inverse of the complex compliance, $E(\omega) = \sigma(\omega, t)/\varepsilon(\omega, t)$.

7.5 Spectrum Representation

The Prony series (7.26) provides a physical interpretation of polymer behavior as a series of Maxwell models, each with its own decay time. In the limit, a real polymer has an infinite number of such models [51], so that

$$E(t) - E_\infty = \int_{-\infty}^{\infty} H(\theta) \exp(-t/\theta) d \ln \theta = \int_0^{\infty} \frac{H(\theta)}{\theta} \exp(-t/\theta) d\theta \tag{7.58}$$

where $H(\theta)$ is the relaxation spectrum [52]. In terms of compliance, we have

$$D(t) - D_0 = \frac{t}{\eta} + \int_{-\infty}^{\infty} \frac{L(\theta)}{\theta} [1 - \exp(-t/\theta)] d\theta \tag{7.59}$$

where $L(\theta)$ is the retardation spectrum [52], D_0 is the elastic compliance, η is the asymptotic viscosity of liquids, with $\eta \to \infty$ for solids (see also [53]).

7.6 Micromechanics of Viscoelastic Composites

7.6.1 One-Dimensional Case

Recall the constitutive equations (7.51) in the Carson domain. By the correspondence principle, all equations of micromechanics for elastic materials are valid in the Carson domain for linear viscoelastic materials. For example, the Reuss micromechanical model assumes uniform identical strain in the matrix and fiber (see discussion on page 216). Therefore, the stiffness of the composite **C** is a linear combination of the stiffness of the constituents (fiber and matrix) weighed by their respective volume fractions V_m, V_f

$$\mathbf{C} = V_m \mathbf{C}^m + V_f \mathbf{C}^f \tag{7.60}$$

with $\mathbf{A}^m = \mathbf{A}^f = \mathbf{I}$ in (6.1). Taking into account the correspondence principle for a viscoelastic material (Section 7.3), it is possible to write the stiffness tensor in the Carson domain by analogy with (7.60) simply as

$$\hat{\mathbf{C}}(s) = V_m \hat{\mathbf{C}}^m + V_f \hat{\mathbf{C}}^f \tag{7.61}$$

From it, the stiffness tensor in the Laplace domain is (see (7.47))

$$\mathbf{C}(s) = \frac{1}{s}\hat{\mathbf{C}}(s) \tag{7.62}$$

Finally, the stiffness tensor in the time domain is obtained by finding the inverse Laplace transform (7.48) as

$$\mathbf{C}(t) = L^{-1}[\mathbf{C}(s)] \tag{7.63}$$

Example 7.3 *Derive the transverse compliance $D_2(t)$ in the time domain for a unidirectional composite with elastic fibers and a viscoelastic matrix represented by $D_m = 1/E_m + t/\eta_m$. Plot D_f, $D_m(t)$, and $D_2(t)$ for $0 < t < 0.1$, $E_f = 10$, $V_f = 0.5$, $E_m = 5$, $\eta_m = 0.05$. Use the Reuss model and discuss the results.*

Solution to Example 7.3 *The elastic behavior of the fiber and viscoelastic behavior of the matrix are defined as follows:*

Fiber (elastic): $E_f = constant \rightarrow D_f = \dfrac{1}{E_f}$

Matrix (Maxwell model (7.12) with $E_m = E_0$, $\eta_m = \tau E_0$*):* $\dfrac{1}{E_m} = \dfrac{1}{E_m} + \dfrac{t}{\eta_m}$

Take the Laplace transform,

$D_f(s) = \dfrac{1}{sE_f}$ *because* $\dfrac{1}{E_f}$ *is constant.*

$D_m(s) = \dfrac{1}{sE_m} + \dfrac{1}{s^2\eta_m}$

Then, the Carson transform is

$\hat{D}_f(s) = s\ D_f(s) = \dfrac{1}{E_f}$

$\hat{D}_m(s) = s\ D_m(s) = \dfrac{1}{E_m} + \dfrac{t}{s\eta_m}$

Using the Reuss model (page 216) to compute the composite behavior

$\hat{D}_2 = V_f\hat{D}_f + V_m\hat{D}_m$

$\hat{D}_2 = V_f\dfrac{1}{E_f} + V_m\left(\dfrac{1}{E_m} + \dfrac{1}{s\eta_m}\right)$

Back to Laplace domain

$D_2(s) = \dfrac{V_f}{sE_f} + \dfrac{V_m}{sE_m} + \dfrac{V_m}{s^2\eta_m}$

Back to the time domain (inverse Laplace)

$D_2(t) = L^{-1}(D_2(s)) = \dfrac{V_f}{E_f} + \dfrac{V_m(E_m t + \eta_m)}{E_m\eta_m}$

To make a plot, take $E_f = 10$, $V_f = 0.5$, $E_m = 5$, $\eta_m = 0.05$, which results in

$$\begin{aligned} &D_f = 0.1 = 1/10 \\ &D_m(t) = 0.2 + 20t \\ &D_2(t) = 0.15 + 10t \end{aligned}$$

Since $V_f = 0.5$, the initial compliance is halfway between those of the fiber and the matrix. The elastic fiber has constant compliance. The creep rate of the composite $1/\eta_c$ is 1/2 of the creep rate of the matrix $1/\eta_m$.

7.6.2 Three-Dimensional Case

The constitutive equation for an elastic, isotropic material (1.78) can be written in terms of just two material parameters λ and $\mu = G$ as

$$\boldsymbol{\sigma} = (\lambda \mathbf{I}^{(2)} \otimes \mathbf{I}^{(2)} + 2\mu \mathbf{I}^{(4)}) : \boldsymbol{\varepsilon} \tag{7.64}$$

where $\mathbf{I}^{(2)}$ and $\mathbf{I}^{(4)}$ are the second- and fourth-order identity tensors[1] (see Appendix A). The constitutive equation of isotropic viscoelastic materials can be written in terms of the viscoelastic Lamé constants $\lambda(s)$ and $\mu(s)$ as follows [53]

$$\boldsymbol{\sigma}(t) = \int_0^t \lambda(t-\theta) \mathbf{I}^{(2)} \otimes \mathbf{I}^{(2)} : \ \dot{\boldsymbol{\varepsilon}}(\theta) d\theta + \int_0^t 2\mu(t-\theta) \mathbf{I}^{(4)} : \dot{\boldsymbol{\varepsilon}}(\theta) d\theta \tag{7.65}$$

Using the convolution theorem (Table 7.1), the Laplace transform of (7.65) is

$$\boldsymbol{\sigma}(s) = s\lambda(s) \mathbf{I}^{(2)} \otimes \mathbf{I}^{(2)} : \boldsymbol{\varepsilon}(s) + s \ 2\mu(s) \mathbf{I}^{(4)} : \boldsymbol{\varepsilon}(s) \tag{7.66}$$

or in terms of the Carson transform

$$\hat{\boldsymbol{\sigma}}(s) = \hat{\mathbf{C}}(s) : \hat{\boldsymbol{\varepsilon}}(s) \tag{7.67}$$

Assuming that the fiber is elastic, and the matrix is viscoelastic, the latter represented with a Maxwell model

$$D_m(t) = 1/E_m + t/\eta_m \tag{7.68}$$

the Carson transform is

$$\hat{D}_m = 1/E_m + 1/s\eta_m = \frac{E_m + s\eta_m}{s\eta_m E_m} \tag{7.69}$$

Using the correspondence principle yields

$$\hat{E}_m = 1/\hat{D}_m = \frac{s\eta_m E_m}{E_m + s\eta_m} = \frac{sE_m}{E_m/\eta_m + s} \tag{7.70}$$

[1] Tensors are indicated by boldface type, or by their components using index notation.

Using (1.75) and assuming the Poisson's ratio ν_m of the matrix to be constant, the Lamé constant of the matrix in the Carson domain is

$$\hat{\lambda}_m = \frac{\hat{E}_m \nu_m}{(1+\nu_m)(1-2\nu_m)} \tag{7.71}$$

and the shear modulus of the matrix is

$$\hat{\mu}_m = \frac{\hat{E}_m}{2(1+\nu_m)} \tag{7.72}$$

Barbero and Luciano [39] used the the Fourier expansion method to get the components of the relaxation tensor in the Carson domain for a composite with cylindrical fibers arranged in a square array with fiber volume fraction V_f. The elastic, transversely isotropic fibers are represented by the transversely isotropic stiffness tensor $\mathbf{C}'$ defined by (1.70,1.92) in terms of fiber properties in the axial and transverse (radial) directions E_A, E_T, G_A, G_T, and ν_T. Defining the matrix properties in the Laplace $\widetilde{()}$ and Carson domain $\widehat{()}$ as $\widehat{\lambda}_m = s\widetilde{\lambda}_m(s)$ and $\widehat{\mu}_m = s\widetilde{\mu}_m(s)$, the components of the relaxation tensor of the composite in the Carson domain $\widehat{\mathbf{L}}^*$ become [39]

$$\widehat{L}^*_{11}(s) = \widehat{\lambda}_m + 2\,\widehat{\mu}_m - V_f\,\left(-a_4^2 + a_3^2\right) \left(-\frac{\left(2\,\widehat{\mu}_m + 2\,\widehat{\lambda}_m - C'_{33} - C'_{23}\right)\left(a_4^2 - a_3^2\right)}{a_1} + \frac{2\left(a_4 - a_3\right)\left(\widehat{\lambda}_m - C'_{12}\right)^2}{a_1^2} \right)^{-1}$$

$$\widehat{L}^*_{12}(s) = \widehat{\lambda}_m + V_f \left(\frac{\left(\widehat{\lambda}_m - C'_{12}\right)\left(a_4 - a_3\right)}{a_1} \right) \left(\frac{\left(2\,\widehat{\mu}_m + 2\,\widehat{\lambda}_m - C'_{33} - C'_{23}\right)\left(a_3^2 - a_4^2\right)}{a_1} + \frac{2\left(a_4 - a_3\right)\left(\widehat{\lambda}_m - C'_{12}\right)^2}{a_1^2} \right)^{-1}$$

$$\widehat{L}^*_{22}(s) = \widehat{\lambda}_m + 2\,\widehat{\mu}_m - V_f \left(\frac{\left(2\,\widehat{\mu}_m + 2\,\widehat{\lambda}_m - C'_{33} - C'_{23}\right)a_3}{a_1} - \frac{\left(\widehat{\lambda}_m - C'_{12}\right)^2}{a_1^2} \right) \left(\frac{\left(2\,\widehat{\mu}_m + 2\,\widehat{\lambda}_m - C'_{33} - C'_{23}\right)\left(a_3^2 - a_4^2\right)}{a_1} + \frac{2\left(a_4 - a_3\right)\left(\widehat{\lambda}_m - C'_{12}\right)^2}{a_1^2} \right)^{-1}$$

$$\begin{aligned}\widehat{L}^*_{23}(s) = \widehat{\lambda}_m + V_f &\left(\frac{\left(2\,\widehat{\mu}_m + 2\,\widehat{\lambda}_m - C'_{33} - C'_{23}\right) a_4}{a_1} - \frac{\left(\widehat{\lambda}_m - C'_{12}\right)^2}{a_1^2} \right) \\ &\left(\frac{\left(2\,\widehat{\mu}_m + 2\,\widehat{\lambda}_m - C'_{33} - C'_{23}\right)(a_3^2 - a_4^2)}{a_1} + \frac{2\,(a_4 - a_3)\left(\widehat{\lambda}_m - C'_{12}\right)^2}{a_1^2} \right)^{-1}\end{aligned}$$

$$\widehat{L}^*_{44}(s) = \widehat{\mu}_m - V_f \left(\frac{2}{2\,\widehat{\mu}_m - C'_{22} + C'_{23}} - \left(2\,S_3 - \frac{4\,S_7}{2 - 2\,\nu_m} \right) \widehat{\mu}_m^{-1} \right)^{-1}$$

$$\widehat{L}^*_{66}(s) = \widehat{\mu}_m - V_f \left(\left(\widehat{\mu}_m - C'_{66} \right)^{-1} - \frac{S_3}{\widehat{\mu}_m} \right)^{-1} \tag{7.73}$$

where

$$\begin{aligned}a_1 = {} & 4\,\widehat{\mu}_m^2 - 2\,\widehat{\mu}_m\,C'_{33} + 6\,\widehat{\lambda}_m\,\widehat{\mu}_m - 2\,C'_{11}\,\widehat{\mu}_m - 2\,\widehat{\mu}_m\,C'_{23} + C'_{23}\,C'_{11} + 4\,\widehat{\lambda}_m\,C'_{12} \\ & - 2\,{C'_{12}}^2 - \widehat{\lambda}_m\,C'_{33} - 2\,C'_{11}\,\widehat{\lambda}_m + C'_{11}\,C'_{33} - \widehat{\lambda}_m\,C'_{23}\end{aligned}$$

$$\begin{aligned}a_2 = {} & 8\,\widehat{\mu}_m^3 - 8\,\widehat{\mu}_m^2 C'_{33} + 12\,\widehat{\mu}_m^2 \widehat{\lambda}_m - 4\,\widehat{\mu}_m^2 C'_{11} \\ & - 2\,\widehat{\mu}_m\,{C'_{23}}^2 + 4\,\widehat{\mu}_m\,\widehat{\lambda}_m\,C'_{23} + 4\,\widehat{\mu}_m\,C'_{11}\,C'_{33} \\ & - 8\,\widehat{\mu}_m\,\widehat{\lambda}_m\,C'_{33} - 4\,\widehat{\mu}_m\,{C'_{12}}^2 + 2\,\widehat{\mu}_m\,{C'_{33}}^2 - 4\,\widehat{\mu}_m\,C'_{11}\,\widehat{\lambda}_m + 8\,\widehat{\mu}_m\,\widehat{\lambda}_m\,C'_{12} \\ & + 2\,\widehat{\lambda}_m\,C'_{11}\,C'_{33} + 4\,C'_{12}\,C'_{23}\,\widehat{\lambda}_m - 4\,C'_{12}\,C'_{33}\,\widehat{\lambda}_m - 2\,\widehat{\lambda}_m\,C'_{11}\,C'_{23} \\ & - 2\,C'_{23}\,{C'_{12}}^2 + {C'_{23}}^2 C'_{11} + 2\,C'_{33}\,{C'_{12}}^2 - C'_{11}\,{C'_{33}}^2 + \widehat{\lambda}_m\,{C'_{33}}^2 - \widehat{\lambda}_m\,{C'_{23}}^2\end{aligned}$$

$$\begin{aligned}a_3 = {} & \frac{4\,\widehat{\mu}_m^2 + 4\,\widehat{\lambda}_m\,\widehat{\mu}_m - 2\,C'_{11}\,\widehat{\mu}_m - 2\,\widehat{\mu}_m\,C'_{33} - C'_{11}\,\widehat{\lambda}_m - \widehat{\lambda}_m\,C'_{33} - {C'_{12}}^2}{a_2} \\ & + \frac{C'_{11}\,C'_{33} + 2\,\widehat{\lambda}_m\,C'_{12}}{a_2} - \frac{S_3 - \dfrac{S_6}{2 - 2\nu_m}}{\widehat{\mu}_m}\end{aligned}$$

$$\begin{aligned}a_4 = {} & -\frac{-2\,\widehat{\mu}_m\,C'_{23} + 2\,\widehat{\lambda}_m\,\widehat{\mu}_m - \widehat{\lambda}_m\,C'_{23} - C'_{11}\,\widehat{\lambda}_m - {C'_{12}}^2 + 2\,\widehat{\lambda}_m\,C'_{12} + C'_{11}\,C'_{23}}{a_2} \\ & + \frac{S_7}{\widehat{\mu}_m\,(2 - 2\,\nu_m)}\end{aligned} \tag{7.74}$$

The coefficients S_3, S_6, S_7 account for the geometry of the microstructure, including the geometry of the inclusions and their geometrical arrangement [36]. For cylindrical fibers arranged in a square array [37] we have

$$\begin{aligned} S_3 &= 0.49247 - 0.47603V_f - 0.02748V_f^2 \\ S_6 &= 0.36844 - 0.14944V_f - 0.27152V_f^2 \\ S_7 &= 0.12346 - 0.32035V_f + 0.23517V_f^2 \end{aligned} \tag{7.75}$$

Note that (7.73) yield six independent components of the relaxation tensor. This is because (7.73) represent a composite with microstructure arranged in a square array. If the microstructure is random (Figure 1.12), the composite is transversely isotropic (Section 1.12.4) and only five components of the relaxation tensor are independent. When the axis x_1 is the axis of transverse isotropy for the composite, the averaging procedure (6.7) yields the relaxation tensor with transverse isotropy as

$$\begin{aligned} \hat{C}_{11} &= \hat{L}^*_{11} \\ \hat{C}_{12} &= \hat{L}^*_{12} \\ \hat{C}_{22} &= \frac{3}{4}\hat{L}^*_{22} + \frac{1}{4}\hat{L}^*_{23} + \frac{1}{2}\hat{L}^*_{44} \\ \hat{C}_{23} &= \frac{1}{4}\hat{L}^*_{22} + \frac{3}{4}\hat{L}^*_{23} - \frac{1}{2}\hat{L}^*_{44} \\ \hat{C}_{66} &= \hat{L}^*_{66} \end{aligned} \tag{7.76}$$

where the remaining coefficients are found using (1.70) due to transverse isotropy of the material. For example, $\hat{C}_{44} = (\hat{C}_{22} - \hat{C}_{23})/2$. This completes the derivation of the relaxation tensor $\hat{\mathbf{C}}_{ij} = s\mathbf{C}_{ij}(s)$ in the Carson domain. Next, the inverse Laplace transform of each coefficient yields the coefficients of the stiffness tensor in the time domain as

$$\mathbf{C}_{ij}(t) = L^{-1}\left[\frac{1}{s}\hat{\mathbf{C}}_{ij}\right] \tag{7.77}$$

A *MATLAB* code based on [50] is available in [5] to perform the inverse Laplace numerically. Another algorithm is provided in [8, Appendix D].

Example 7.4 *Consider a composite made with 60% by volume of transversally isotropic fibers with axial properties* $E_A = 168.4$ *GPa*, $G_A = 44.1$ *GPa*, $\nu_A = 0.443$, *and transverse properties* $E_T = 24.8$ *GPa and* $\nu_T = 0.005$. *The epoxy matrix is represented by a Maxwell model (7.12) with* $E_0 = 4.08$ *GPa*, $\tau = 39.17$ *min and* $\nu_m = 0.311$. *Plot the relaxation* $E_2(t)$ *of the composite as a function of time for* $0 < t < 100$ *minutes, compared to the elastic value of the transverse modulus* E_2.

Solution to Example 7.4 *This example has been solved using* MATLAB. *The elastic and viscoelastic values of the transverse modulus* E_2 *are shown in Figure 7.4. The calculation procedure is explained next:*

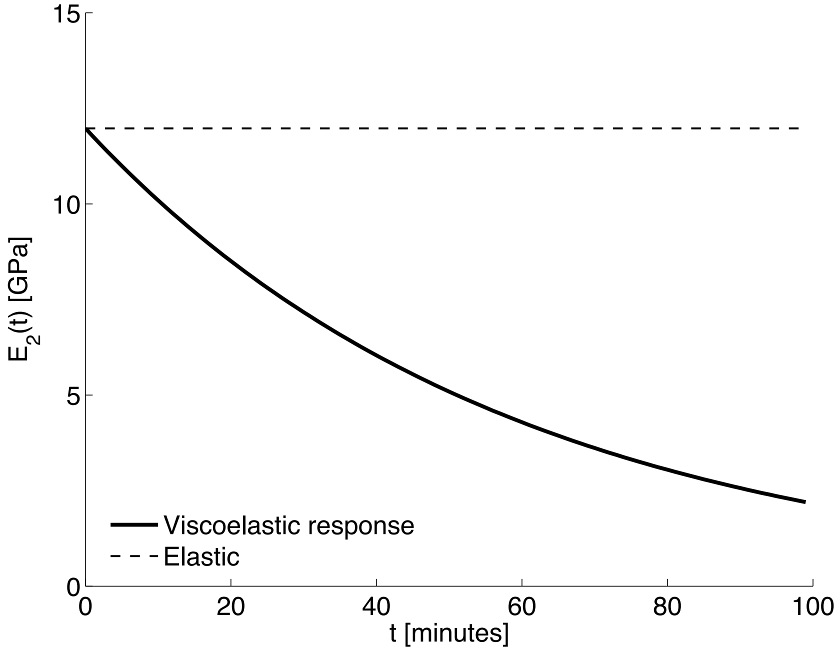


Fig. 7.4: Elastic and viscoelastic values of the transverse modulus E_2.

- *Program the equations of Section 7.6.2 and use them to calculate the elastic values of the composite's elastic properties such as E_2. These equations have been implemented in* `PMMViscoMatrix.m`*.*

- *Replace the elastic modulus of the matrix E_0 by the Maxwell model for the matrix Eq. (7.15) in the Carson domain $\hat{E}_0$ (see* `PMMViscoMatrix.m`*), as follows:*

  i. *The output from the portion of the code implementing (7.73-7.76) are equations for the relaxation moduli in terms of s in the Carson domain. Note that it is necessary to declare the variable s as symbolic.*

  ii. *Divide them by s to go back to the Laplace domain.*

  iii. *Back transform to the time domain using the function* `invlapFEAcomp`*, which is derived from [50].*

  iv. *Finally, fit the numerical values of $E_2(t)$ with a viscoelastic model equation. Usually it is convenient to use the same model equation for the composite relaxation as that used for the matrix relaxation; in this case the Maxwell model. This step is implemented in* `fitfunFEAcomp.m`

The MATLAB *codes* `PMMViscoMatrix.m`*,* `invlapFEAcomp.m`*, and* `fitfunFEAcomp.m` *are available in [5]. The results are shown in Figure 7.4. The complete set of Maxwell parameters for the composite are calculated in Example 7.7, p. 272.*

7.7 Macromechanics of Viscoelastic Composites

7.7.1 Balanced Symmetric Laminates

The in-plane viscoelastic behavior of a balanced symmetric laminate can be obtained using the procedure in Section 1.15 (Apparent Laminate Properties), but in the Carson domain. Start with the stiffness of the laminas (7.76) in lamina coordinates, in the Carson domain. Rotate each matrix to laminate coordinates. Then, average them using (1.102). Using (1.105), find the laminate engineering properties in the Carson domain and divide by s to go back to the Laplace domain. Finally, take the inverse Laplace transform to find the laminate stiffness in the time domain. Then, fit them with a model equation as it is done in Example 7.4, p. 267.

7.7.2 General Laminates

Thanks to the correspondence principle, the stress-resultant vs. strain-curvature equations from classical lamination theory (CLT, see Chapter 3) are valid for linearly viscoelastic laminated composites in the Carson domain. The A, B, D, H matrices of a laminate in the Carson domain can be computed by using the equations from first-order shear deformation theory (FSDT, Section 3.1.1). This methodology was used in [54].

7.8 FEA of Viscoelastic Composites

Most commercial codes have implemented viscoelasticity (creep) for isotropic materials. This is a severe limitation for users interested in the analysis of viscoelastic behavior of polymer matrix composites.

However, it is possible to take advantage of the user programmable features of commercial software in order to implement the formulations presented in this chapter. This is relatively easy because the approach used in this chapter is not stress dependent, but a linear viscoelastic approach, and its implementation is not complicated. A UMAT subroutine is used in Example 7.7 (p. 272) to implement the viscoelastic formulation.

Example 7.5 *Compute the creep response of a* $[0/90_8]_s$ *laminate. The thickness of each layer is* $t_k = 1.25\ mm$*. The laminate width is* $2b = 20\ mm$ *and it has a length* $2L = 80\ mm$*. Load the sample with a uniform stress* σ_x *by using coupling constraints on the free end at* $x = L$ *and a load* $P = 1000\ N$*. Use solid elements on each layer and symmetry conditions to model 1/8 of the laminate. Both laminas are isotropic with the properties* $E_0 = 11975\ MPa$*,* $\nu_{23} = 0.1886$*,* $g_1 = 0.999$ *(Maxwell),* $k_1 = 0.999$ *(constant Poisson's), and* $\tau = 58.424\ min$*.*

Solution to Example 7.5 *Most of the model can be obtained by modifying the model of Example 5.2 (p. 199) as illustrated in the following pseudo code:*

i. Retrieve the `cae` *file of Example 5.2*

```
Menu:   File, Set Work Directory, [C:\SIMULIA\User\Ex_7.5], OK
Menu:   File, Open, [C:\SIMULIA\User\Ex_5.2\Ex_5.2.cae], OK
Menu:   File, Save As, [C:\SIMULIA\User\Ex_7.5\Ex_7.5.cae], OK
```

ii. *Modifying the part*

```
Module: Part
    # change thickness of 90-deg layer editing the section solid extrude
    # on the .mdb tree, # expand: Parts(1), Part-1, Features
    # right-click: Solid extrude, Edit, Depth [11.25], OK
    # move the partition plane
    # right-click: Datum plane-1, Edit, Offset [10], OK
```

iii. *Modifying materials and section properties. The material defined as* `Material-1` *in Example 5.2 must be removed because visco-analysis in* Abaqus *only accepts isotropic materials.*

```
Module: Property
Menu:   Material, Manager
    # select: Material-1, Delete, Yes
    # create an isotropic visco-elastic material
    Create, Name [iso-visco], Mechanical, Elasticity, Elastic
    Type: Isotropic, [11975, 0.1866]
    Moduli time scale: Instantaneous
    # viscous part
    Mechanical, Elasticity, Viscoelastic, Domain: Time, Time: Prony
    g_1 [0.999], k_1 [0.999], tau_1 [58.242], OK
    # close Material Manager pop-up window
Menu:   Section, Edit, Section-1, Material: iso-visco, OK
```

iv. *Modifying analysis step*

```
Module: Step
Menu:   Step, Manager
    # select: Step-1, Delete, Yes, # this also deletes the BC applied
    Create, Name [Step-1], Procedure type: General, Visco, Cont
    Tab: Basic, Time period [150]
    Tab: Incrementation, Type: Fixed, Max number of increments [200]
    Increment size [1], OK, # close Step Manager pop-up window
```

v. *Defining constraints and interactions*

```
Module: Interaction
    # make surface X=40 into a rigid surface to apply a force
Menu:   Tools, Reference Point, [40,10,11.25], X # RP-1
Menu:   Constraint, Create
    Name [Constraint-1], Type: Coupling, Cont
    # pick: RP-1 # constant control point
    Surface, # pick the surfaces @ X=40, Done
    Constrained degrees of freedom: U1 # uncheckmark: all but U1, OK
```

vi. *Creating a set to request History Output*

```
Module: Step, Model: Model-1, Step: Step-1 # Step-1 must be selected
Menu:   Tools, Set, Create
    Name [Set-1], Cont, # pick: RP-1, Done
Menu:   Output, History Output Requests, Edit, H-Output-1
```

Table 7.3: Results Example 7.5.

Time [min]	U1 [mm]	ϵ_x [%]	σ_x [MPa]
0	0.0000	0.0000	0.0000
1	0.0299	0.0749	8.8889
150	0.1057	0.2643	8.8889

```
    Domain: Set : Set-1, Output Variables: Select from list below
    # expand: Stresses, #checkmark: S # make sure all terms are selected
    # expand: Strains,  #checkmark: E # make sure all terms are selected
    # expand: Displacement/Velocity/Acceleration, # checkmark: U, OK
```

vii. *Adding loads and BC*

```
Module: Load
    # restrain rotations UR2, UR3, of the reference point
Menu:   BC, Create
    Name [BC-RP-1], Step: Initial, Type: Disp/Rota, Cont
    # pick: RP-1, Done, # checkmark: UR2, UR3, OK
    # apply load at the reference point
Menu:   Load, Create
    Step: Step-1, Mechanical, Concentrated force, Cont
    # pick: RP-1, Done, CF1 [1000.0], OK
```

viii. *Remeshing the model*

```
Module: Mesh
Menu:   Mesh, Instance, Yes
```

ix. *Solving and visualizing the results*

```
Module: Job
Menu:   Job, Manager
    Create, Cont, OK, Submit, # when Completed, Results
Module: Visualization
    # plot displacement vs. time
Menu:   Result, History Output, U1, Plot, # close pop-up window
Menu:   Plot, Contours, On Deformed Shape
    Toolbox: Field Output, Primary, U, U1
Menu:   Result, Step/Frame, Index: 0, Apply
Menu:   Result, Step/Frame, Index: 1, Apply
Menu:   Result, Step/Frame, Index: 150, Apply
    # analogously visualize the results of E11 and S11
```

The results are shown in Table 7.3. Note that the contour plots of stress and strain display results in the local coordinate systems that were assigned in Example 5.2, and thus the results shown in Table 7.3 are extracted from the outer lamina. This has been done on purpose to emphasize that contour plots display results in the local coordinate system defined by CSYS for each lamina.

Example 7.6 *Compute the relaxation response of the laminate in Example 7.5. Load the sample with a uniform strain $\epsilon_x = 0.1$ by using coupling constraints on the free end at $x = L$ and imposing a displacement $U1 = 4$ mm.*

Solution to Example 7.6 *Most of the model can be obtained by modifying the model of Example 7.5 as illustrated in the following pseudo code. Note how the initial displacement is applied over a short period of time (1 min) using automatic incrementation. Then, a second step is set up to compute the relaxation over 150 min with fixed intervals to facilitate data interpretation.*

i. Retrieve the `cae` *file of Example 7.5*

```
Menu:   Set Work Directory, [C:\SIMULIA\User\Ex_7.6], OK
Menu:   File, Open, [C:\SIMULIA\User\Ex_7.5\Ex_7.5.cae], OK
Menu:   File, Save As, [C:\SIMULIA\User\Ex_7.6\Ex_7.6.cae], OK
```

ii. Modifying analysis step

```
Module: Step
Menu:   Step, Edit, Step-1
    Tab: Basic, Time period [1]
    Tab: Incrementation, Type: Automatic
    Creep/swelling/viscoelastic strain error tolerance: [1E-6], OK
Menu:   Step, Create
    Name [Step-2], Procedure type: General, Visco, Cont
    Tab: Basic, Time period [150]
    Tab: Incrementation, Type: Fixed, Max number of increments [200]
    Increment Size [1], OK
```

iii. Modifying the applied loads and BC

```
Module: Load
    # delete the load at RP-1
Menu:   Load, Delete, Load-1, Yes
    # specify a displacement at RP-1
Menu:   BC, Create, Step: Step-1, Disp/Rota, Cont
    # pick: RP-1, Done, # checkmark: U1 [4.0], OK
```

iv. Solving and visualizing the results

```
Module: Job
    # create new Job. Do not overwrite the results of the creep case
Menu:   Job, Manager
    Create, Cont, OK, Submit, # when Completed, Results
Module: Visualization
    # plot displacement vs. time
Menu:   Result, History Output, U1, Plot, # close pop-up window
    # visualize the results at different times as in Example 7.5
```

The results are shown in Table 7.4.

Example 7.7 *[8, Example 7.5] Compute the relaxation response of a $[0/90_8]_s$ laminate. The thickness of each layer is $t_k = 1.25$ mm. The laminate width is $2b = 20$ mm and its length is $2L = 40$ mm. Load the sample with a uniform strain $\epsilon_x = 0.1$ by applying a uniform displacement at $x = L$. Use solid elements on each layer and symmetry conditions. Plot the laminate stiffness $E_x(t)$ for $0 > t > 150$ minutes. Use the lamina material properties given in Table 7.5, which were computed with the procedure used in Example 7.4, p. 267.*

Table 7.4: Results Example 7.6

Total Time [min]	Step Time [min]	Step	U1 [mm]	ϵ_x	σ_x [MPa]
0	0	Step-1	0.00	0.00	0.00
1	1	Step-1	4.00	0.10	1187.00
1	0	Step-2	4.00	0.10	1187.00
3	2	Step-2	4.00	0.10	1147.00
11	10	Step-2	4.00	0.10	1001.00
151	150	Step-2	4.00	0.10	91.48

Table 7.5: Lamina viscoelastic properties for Example 7.7.

Young's Moduli	**Shear Moduli**	**Poisson's Ratio**
$(E_1)_0 = 102417$ MPa	$(G_{12})_0 = (G_{13})_0 = 5553.8$ MPa	$\nu_{12} = \nu_{13} = 0.4010$
$\tau_1 = 16551$ min	$\tau_{12} = \tau_{13} = 44.379$ min	
$(E_2)_0 = (E_3)_0 = 11975$ MPa	$(G_{23})_0 = 5037.3$ MPa	$\nu_{23} = 0.1886$
$\tau_2 = \tau_3 = 58.424$ min	$\tau_{23} = 54.445$ min	

Solution to Example 7.7 *The Maxwell parameters of the lamina are shown in Table 7.5. Using a UMAT subroutine in* Abaqus*, it is possible to implement the constitutive equation of an orthotropic material with the following time-dependent properties:*

$E_1(t) = (E_1)_0 \exp(-t/\tau_1)$; $E_2(t) = E_3(t) = (E_2)_0 \exp(-t/\tau_2)$ *and*

$G_{12}(t) = G_{13}(t) = (G_{12})_0 \exp(-t/\tau_{12})$; $G_{23}(t) = (G_{23})_0 \exp(-t/\tau_{23})$.

This subroutine is available in [5, umat3dvisco.for]. The rest of the model can be constructed using a procedure similar to that used in Example 7.6, p. 271.

i. *Retrieving* `cae` *file of Example 7.6 and setting* `Work Directory`

```
Menu:   Set Work Directory, [C:\SIMULIA\User\Ex_7.7], OK
Menu:   File, Open, [C:\SIMULIA\User\Ex_7.6\Ex_7.6.cae], OK
Menu:   File, Save As, [C:\SIMULIA\User\Ex_7.7\Ex_7.7.cae], OK
```

ii. *Creating a material to be used with the UMAT*

```
Module: Property
Menu:   Material, Create
    Name [user], General, User Material, Type: Mechanical
    [102417 11975 0.401 0.1886 5553.8 5037.3 16551 58.424 44.379
     54.445]
    # E1o, E2o, nu12o, nu23o, G12o, G23o, tau1, tau2, tau12, tau23
    # entered as a column vector
    General, Depvar,
    Number of solution-dependent state variables [6], OK
```

iii. *Modifying the analysis steps*

```
Module: Step
Menu:   Step, Manager
    # pick: Step-1, Edit, Tab: Basic, Time period [0.001]
    Tab: Incrementation, Type: Automatic, Error tolerance [1E-6], OK
    # leave Step-2 as it is, # close Step Manager pop-up window
```

```
Step:   Step-1, # selected above WS
Menu:   Output, Field Output Requests, Edit, F-Output-1
        Output Variables: Edit variables [S,E,U,RF,SDV], OK
Menu:   Output, History Output Requests, Edit, H-Output-1
        Output Variables: Edit variables [S11,E11,U1,RF1,SDV], OK
```

iv. *Remeshing the model with quadratic elements to avoid specifying hourglass stiffness*

```
Module: Mesh
Menu:   Mesh, Element Type, # select all, Done
    Family: 3D Stress, Geometric Order: Quadratic
    # checkmark: Reduced integration, OK
Menu:   Mesh, Instance, Yes
```

v. *Solving and visualizing the results*

```
Module: Job
    # create a third Job
    # but do not to overwrite the results of previous one
Menu:   Job, Manager
    Create, Cont, Tab: General,
        User subroutine file [umat3dvisco.for], OK
    Submit, # when Completed, Results
Module: Visualization
    # visualize the reaction force to calculate laminate stiffness
    Toolbox: Field Output, Symbol, RF, RESULTANT
    # to save data to Excel
    # on the Left Tree
    # expand: Output Databases, # expand: Job-4.odb (Job name used)
    # expand: History Output, # right-click: Reaction Force
    Save As, Name [XYData-RF1], OK
    # expand: XYData, # right-click: XYData-RF1, Plot
    # # right-click: XYData-RF1, Edit, # select: all data
    # right-click: Copy, # paste in Excel
    # close pop-up window
```

The results are shown in Figure 7.6.

In Abaqus *an analysis is often divided into steps $s = 1...ns$. Further, each step is often divided into increments $i = 1...ni$.* Abaqus *maintains a record of the (total)* time *elapsed t since the beginning of the analysis, including restart steps. In addition,* Abaqus *maintains a record of the time elapsed since the beginning of the step, called* step time *t_s.*

With reference to Figure 7.5, the step time t_s at the beginning of the current increment (within the current step) is passed to the user subroutines in the variable `TIME(1)`. The (total) time t at the beginning of the current increment is `TIME(2)`. The current time increment ΔT is `DTIME`.

In this way the current step time can be calculated as

$$t_s = \texttt{TIME(1)} + \texttt{DTIME}$$

and the current (total) time can be calculated as

$$t = \texttt{TIME(2)} + \texttt{DTIME}$$

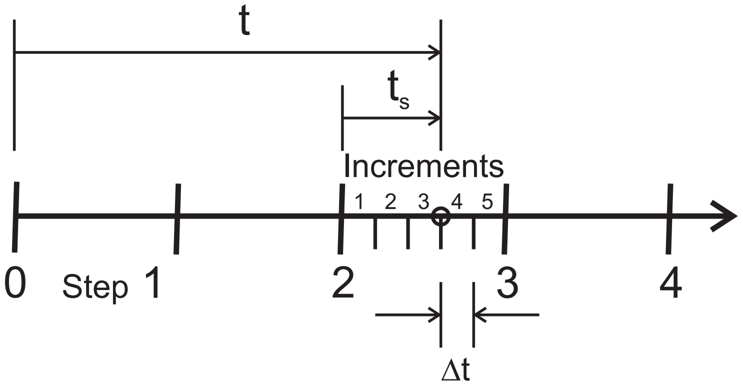


Fig. 7.5: Definition of step time t_s, total time t, and time increment Δt.

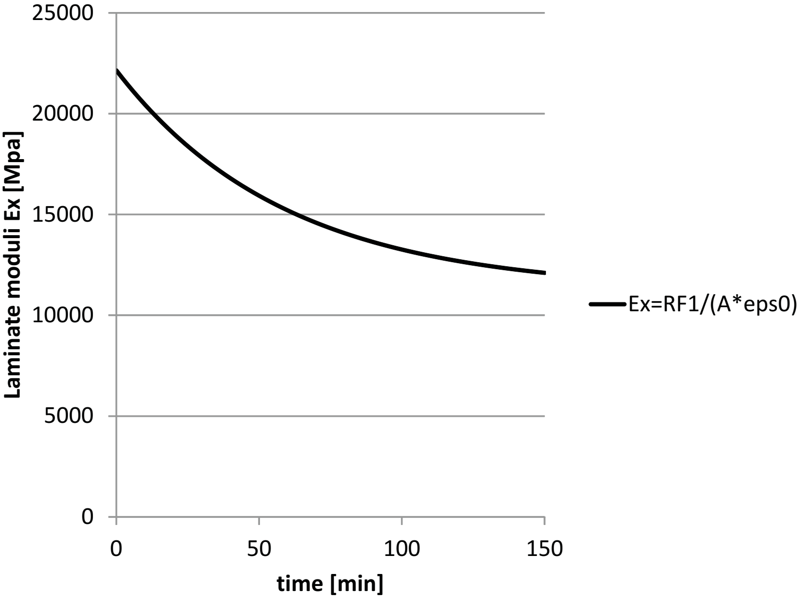


Fig. 7.6: Laminate stiffness $E_x(t)$ calculated in Example 7.7.

Example 7.8 *Consider a composite made with 40% by volume of isotropic graphite fibers with properties $E_f = 168.4$ GPa, $\nu_f = 0.443$ and epoxy matrix represented by a Maxwell model with $E_0 = 4.082$ GPa, $\tau = 39.15$ min, and $\nu_m = 0.311$ (independent of time). Construct a FE micromechanical model using hexagonal microstructure (see Example 6.3, p. 236), subject to shear strain $\gamma_4 = 0.02$ applied suddenly at $t = 0$. Tabulate the average stress σ_4 over the RVE at times $t = 0$, 20, 40, 60, 80, and 100 minutes.*

Solution to Example 7.8 *The fiber can be represented using elastic properties in* Abaqus. *The matrix can be represented using viscoelastic properties in* Abaqus. *The model definition is very similar to Example 6.4, p. 240. There are two ways to solve this example: one is to modify the Python script `Ex_6.4-CE.py`; the other is to modify `Ex_6.4-CE.cae`. The first approach is illustrated next.*

First, copy `PBC_2D.py` and `srecover2D.py` to the local directory for this example. Then, copy `Ex_6.4-CE.py` as `Ex_7.8.py` and open it with Notepad++ for editing. Start by updating the work directory to the current one, `C:\SIMULIA\User\Ex_7.8`, by modifying the corresponding line in `Ex_7.8.py` as follows:

```
os.chdir(r'C:\SIMULIA\User\Ex_7.8')
```

Then, modify the material properties to include all the viscoelastic parameters for the matrix. The elastic moduli are given in TPa because the dimensions are given in micrometers.

```
Ef, nuf = 168.4E-3, 0.443  # TPa
Em, num, tau, g_1, k_1 = 4.082E-3, 0.311, 39.15, 0.999, 0.999 # TPa,,min,,,
```

Then, modify the loading, keeping only $\gamma_4 = 0.02$ (gamma_12 in the coordinate system of the model):

```
strain = [0.000, 0.0, 0.01]    # epsilon_11, epslion_22, gamma_12
```

Next, we need a new definition of the viscoelastic matrix material. The fiber remains as is–elastic. To get the Python script for a viscoelastic material, open Abaqus/*CAE and save the just opened session as `Ex_7.8.cae`. Then, define a viscoelastic material exactly as it was done in Example 7.5, p. 269. Here is the pseudo code detailing the actions to be done in* Abaqus/*CAE:*

```
Module: Property
Menu:   Material, Create
    Name [matrix], Mechanical, Elasticity, Elastic, Type: Isotropic
    Moduli time scale: Instantaneous, Data [4.082E-3, 0.311]
    Mechanical, Elasticity, Viscoelastic, Domain: Time, Time: Prony
    g_1, k_1, tau_1 [0.999, 0.999, 39.15], OK
```

Before saving the model, copy the following lines from `Ex_7.8.rec` to `Ex_7.8.py`, replacing the two lines defining the properties of the matrix material in the section named `# Materials`.

```
mdb.models['Model-1'].Material(name='matrix')
mdb.models['Model-1'].materials['matrix'].Elastic(moduli=INSTANTANEOUS
    , table=((0.004082, 0.311), ))
mdb.models['Model-1'].materials['matrix'].Viscoelastic(domain=TIME,
    table=((0.999, 0.999, 39.15), ), time=PRONY)
```

and immediately replace numerical values by variable names (to parameterize the script), as follows:

```
mdb.models['Model-1'].Material(name='matrix')
mdb.models['Model-1'].materials['matrix'].Elastic(moduli=INSTANTANEOUS
    , table=((Em, num), ))
mdb.models['Model-1'].materials['matrix'].Viscoelastic(domain=TIME,
    table=((g_1, k_1, tau), ), time=PRONY)
```

Now, save the model to reset the `.rec` *file. Now, create a viscoelastic step-1 to apply the strain, and a step-2 to track the ensuing relaxation, exactly as it was done in Example 7.6. Here is the pseudo code detailing the actions to be done in* Abaqus/*CAE:*

```
Module: Step
Menu:   Step, Create
    Name [Step-1], Procedure type: General, Visco, Cont
    Tab: Basic, Time period [0.001]
    Tab: Incrementation, Type: Automatic, viscoelastic tolerance [1E-6], OK
Menu:   Step, Create
    Name [Step-2], Procedure type: General, Visco, Cont
    Tab: Basic, Time period [100]
    Tab: Incrementation, Type: Fixed, Maximum number of increments: [200]
    Increment Size [1], OK
```

Before saving the model, copy the following lines from `Ex_7.8.rec` *to* `Ex_7.8.py`*, replacing the line defining the Step-1 in the section named* `# Step`*.*

```
mdb.models['Model-1'].ViscoStep(cetol=1e-06, initialInc=0.001, maxInc=0.001,
    minInc=1e-08, name='Step-1', previous='Initial', timePeriod=0.001)
mdb.models['Model-1'].ViscoStep(cetol=0.0, initialInc=1.0, maxNumInc=200,
    name='Step-2', previous='Step-1', timeIncrementationMethod=FIXED,
    timePeriod=100.0)
```

Next, update the name of the file to add the BC to the model

```
execfile('C:/SIMULIA/User/Ex_7.8/PBC_2D.py')
```

Next, update the name of the file in

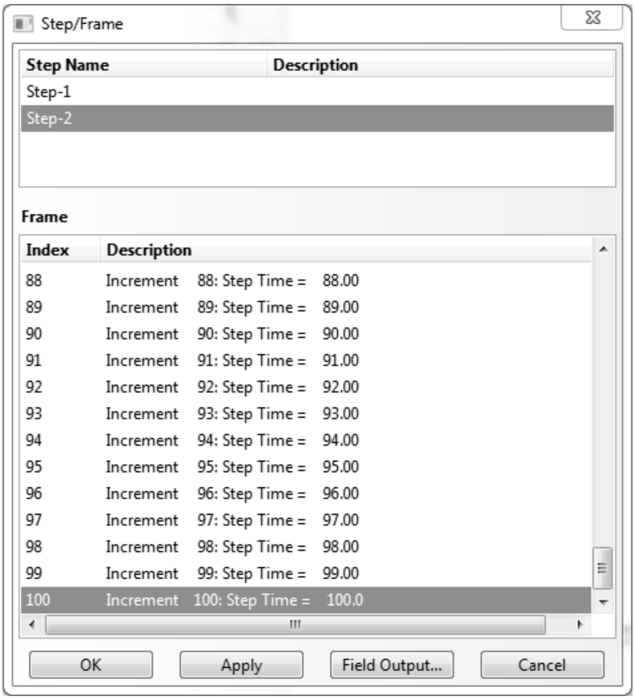
```
mdb.saveAs(pathName='C:/SIMULIA/User/Ex_7.8/Ex_7.8.cae')
```

Next, comment out the following lines

```
# Calculate Stresses and Strains
# execfile('srecover2D.py')
# visualize
# o3 = session.openOdb(name='Job-1.odb')
# session.viewports['Viewport: 1'].setValues(displayedObject=o3)
```

At this point, your updated script should look like the one available in [5, Ex_7.8.py]. Next, reset the Abaqus/*CAE session to start a new model, and then run the script, as follows:*

```
Menu:   File, New Model Database, With Standard/Explicit Model
Menu:   File, Run Script [Ex_7.8.py], OK
```



Step Name	Description
Step-1	
Step-2	

Index	Description
88	Increment 88: Step Time = 88.00
89	Increment 89: Step Time = 89.00
90	Increment 90: Step Time = 90.00
91	Increment 91: Step Time = 91.00
92	Increment 92: Step Time = 92.00
93	Increment 93: Step Time = 93.00
94	Increment 94: Step Time = 94.00
95	Increment 95: Step Time = 95.00
96	Increment 96: Step Time = 96.00
97	Increment 97: Step Time = 97.00
98	Increment 98: Step Time = 98.00
99	Increment 99: Step Time = 99.00
100	Increment 100: Step Time = 100.0

Fig. 7.7: The Step/Frame window is used to select the step and frame (increment) on the output database.

To visualize the results, do the following:

```
Module: Job
Menu:   Job, Results, Job-1
Module: Visualization
Menu:   Plot, Contours, On Deformed Shape
Menu:   Result, Step/Frame
```

Note how the `Menu: Result, Step/Frame` opens the `Step/Frame` window shown in Figure 7.7. Here, you can select the Step and Increment. In this example, `Step-1` is the loading and `Step-2` is the relaxation. Since `Incrementation: Fixed` was used in Step-2, increment numbers correspond to time in minutes. For a time-dependent analysis such as this example, frame and increment are synonymous.

To calculate the average stress $\overline{\sigma}_4$ (S12 in the model) and average strain $\overline{\gamma}_4$ (E12 in the model), modify `srecover2D.py` as follows.

i. Select the step. In this example, step-1 is the loading step and step-2 is the relaxation step. The step is selected in the script as follows:

```
# In Ex. 7.8, Step-2 is the relaxation step
frameRepository = odb.steps['Step-2'].frames;
```

ii. Parameterize the script to select the frame where the increment is stored:

```
# Get the results for frame [i], where i is the increment number
i = 0
```

```
     frameS = frameRepository[i].fieldOutputs['S'].values;
     frameE = frameRepository[i].fieldOutputs['E'].values;
     frameIVOL = frameRepository[i].fieldOutputs['IVOL'].values;
```

The rest of the script is unchanged. The complete script is given below (also available [5, srecover2D.py]).

```
# srecover2D.py modified for Ex. 7.8
from visualization import *
# Open the output data base for the current Job
odb = openOdb(path='Job-1.odb');
myAssembly = odb.rootAssembly;

# Temporary variable to hold the frame repository speeds up the process
# In Ex. 7.8, Step-2 is the relaxation step
frameRepository = odb.steps['Step-2'].frames;

# Get the results for frame [i], where i is the increment number
i = 80
frameS = frameRepository[i].fieldOutputs['S'].values;
frameE = frameRepository[i].fieldOutputs['E'].values;
frameIVOL = frameRepository[i].fieldOutputs['IVOL'].values;

Tot_Vol=0.;          # Total Volume
Tot_Stress=0.;       # Stress Sum
Tot_Strain = 0.;     # Strain Sum

# Calculate Average
for II in range(0,len(frameS)):
     Tot_Vol+=frameIVOL[II].data;
     Tot_Stress+=frameS[II].data * frameIVOL[II].data;
     Tot_Strain+=frameE[II].data * frameIVOL[II].data;

Avg_Stress = Tot_Stress/Tot_Vol;
Avg_Strain = Tot_Strain/Tot_Vol;

# from Abaqus Analysis User's Manual - 1.2.2 Conventions -
# Convention used for stress and strain components
print '2D Abaqus/Standard Stress Tensor Order: 11-22-33-12'
print 'Average stresses Global CSYS: 11-22-33-12';
print Avg_Stress;
print 'Average strain Global CSYS: 11-22-33-12';
print Avg_Strain;
odb.close()
```

The average values at step-1, increment/frame $i = 0$, are strain $\overline{\gamma}_4 = 2.0 \times 10^{-2}$ and average stress $\overline{\sigma}_4 = 6.26 \times 10^{-5}\ TPa$. In fact, the average strain is constant on a relaxation test. By changing the value of Step (Step-1 or Step-2) and frame (i) in the script, all the results can be obtained and tabulated as in Table 7.6.

Using an exponential regression it is possible to calculate the values of initial shear modulus $G_{23}^0 = 3.13\ GPa$ and relaxation time $\tau = 39.97\ min$ that represent the relaxation of the composite in the 23-shear direction using a Maxwell model (see Figure 7.8).

Table 7.6: Average stress σ_4 over time.

Frame: i	0	20	40	60	80	100
Time [min]	0	20	40	60	80	100
Average σ_4 [MPa]	62.6	38.3	23.4	14.4	8.8	5.4

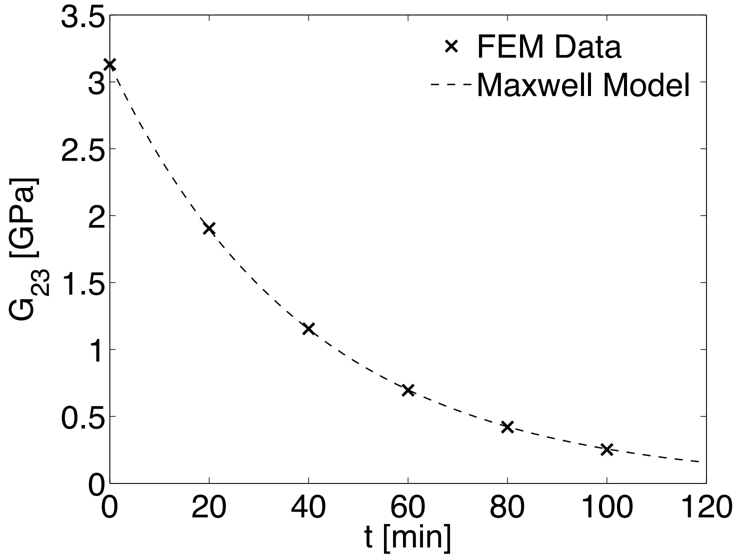


Fig. 7.8: Shear modulus G_{23} as a function of time.

Suggested Problems

Problem 7.1 *Consider a composite made with* 60% *by volume of isotropic fibers with properties* $E_f = 168.4$ *GPa and* $\nu_f = 0.443$, *and epoxy matrix represented by a power law model (7.24) with* $D_0 = 0.222$ *GPa*$^{-1}$, $D_1 = 0.0135$ *(GPa min)*$^{-1}$, $m = 0.17$, *and* $\nu_m = 0.311$. *Plot the relaxation* $C_{22}(t)$ *of the composite as a function of time for* $0 < t < 100$ *minutes. Compare it to the elastic value of the stiffness* C_{22} *of the composite and the elastic stiffness* C_{22} *of the matrix.*

Problem 7.2 *Consider a composite made with 60% by volume of transversely isotropic graphite fibers with properties* $E_A = 168.4$ *GPa,* $E_T = 24.82$ *GPa,* $\nu_A = 0.443$, $\nu_T = 0.005$, $G_A = 44.13$ *GPa, and epoxy matrix represented by a Maxwell model (7.15) with* $E_0 = 4.082$ *GPa,* $\tau = 39.15$ *min, and* $\nu_m = 0.311$. *Plot the relaxation tensor stiffness components* $C(t)$ *of the composite as a function of time for* $0 < t < 100$ *minutes, compared to the elastic stiffness* C *of the composite and the elastic stiffness* C_m *of the matrix.*

Problem 7.3 *Compute the parameters in the Maxwell model for unidirectional lamina (see Section 1.14) of carbon/epoxy material used in Problem 7.2. Plot and compare the elastic and viscoelastic properties:* $E_1(t)$, $E_2(t)$, *and* $G_{12}(t)$. *Show all work in a report.*

Problem 7.4 *Use the user programmable features to implement the Maxwell model constitutive equations for a transversally orthotropic lamina material under plane stress conditions. Using the viscoelastic materials properties obtained in Problem 7.3, compute the response of a* $[\pm 45/90_2]_s$ *laminate. The thickness of each layer is* $t_k = 1.25$ *mm. Load* *the sample with uniform edge load $N_x = 1$ N/mm. Plot the creep compliance $J_x(t)$ of the laminate as a function of time for $0 > t > 300$ minutes.*

Problem 7.5 *Compute the parameters in the Maxwell model for all the nine engineering properties of a $[0/90]_S$ laminate. Each layer is 1.25 mm thick. The material is carbon T300 and Epoxy 934(NR) with $V_f = 0.62$ and layer thickness 1.25 mm. Epoxy is represented by a Maxwell model (7.15) with $E_0 = 4.082$ GPa, $\tau = 39.15$ min, and $\nu_m = 0.311$. Carbon T300 is transversely isotropic with axial modulus $E_A = 202.8$ GPa, transverse modulus $E_T = 25.3$ GPa, $G_A = 44.1$ GPa, $\nu_A = 0.443$, and $\nu_T = 0.005$, where the subscripts A and T indicate the axial and radial (transverse) directions of the fiber, respectively.*

Chapter 8

Continuum Damage Mechanics

Many modes of damage can be observed in composite materials, including matrix cracks, fiber breakage, fiber-matrix de-bonding, and so on. Much work has been done trying to quantify each of these damage modes, their evolution with respect to load, strain, time, number of cycles, etc., and their effect on stiffness, remaining life, etc. *Continuum Damage Mechanics* (CDM) represents all these failure modes by the effect they have on the mesoscale behavior (lamina level) of the material. That is, CDM calculates the degraded moduli of the laminas and laminate in terms of continuum damage variables. Then, either strength or fracture mechanics failure criteria are used to detect damage initiation. Finally, damage evolution is predicted in terms of empirical hardening equations set up in terms of additional parameters, such as the hardening exponent used for metal plasticity. For example, a form of CDM is used in Chapter 10 to represent the degradation of the interface between laminas.

Hardening equations require non-standard experimentation to adjust the additional, empirical parameters. Since the parameters are adjusted to the model, some shortcomings of the model may be masked by the fitting of the parameters. From a thermodynamics point of view, damage variables are the state variables of the formulation, and they are not measurable. This is in contrast to micromechanics of damage models (Chapter 9) and metal plasticity where the state variables, i.e., crack density and plastic strain, are measurable. From a practical point of view, CDM major shortcoming is the need for additional experimentation to determine parameters that are particular to each model. Furthermore, since the state variables are not measurable, the additional parameters need to be adjusted to the model through the loss of stiffness, which may not be sufficiently sensitive to damage [55].

One notable effect of damage is a reduction of stiffness, which can be used to define damage [56]. One-dimensional models are used in Section 8.1 to introduce the concepts. The theoretical formulation for the general three-dimensional case is developed in Sections 8.2-8.4.

8.1 One-Dimensional Damage Mechanics

The development of a one-dimensional damage mechanics solution involves the definition of three major entities: 1) a suitable damage variable, 2) an appropriate damage activation function, and 3) a convenient damage evolution, or kinetic equation.

8.1.1 Damage Variable

Consider a composite rod of nominal area $\widetilde{A}$, unloaded, and free of any damage (Figure 8.1.a). Upon application of a sufficiently large load P, damage appears (Figure 8.1.b). On a macroscopic level, damage can be detected by the loss of stiffness of the material. In CDM, damage is represented by a state variable D, called *damage variable*, which represents the loss of stiffness [56]

$$D = 1 - E/\widetilde{E} \tag{8.1}$$

where $\widetilde{E}$ is the initial (virgin) Young's modulus, and E is the modulus after damage.[1] Earlier work [57] conceptualized damage as the reduction of area due to accumulation of micro-cracks having the same effect as the actual damage

$$D = 1 - A/\widetilde{A} \tag{8.2}$$

where $\widetilde{A}$, A are the initial and remaining cross-sectional areas, respectively. The complement to damage is the integrity [58]

$$\Omega = 1 - D = A/\widetilde{A} \tag{8.3}$$

which can be interpreted as the remaining cross-sectional area ratio, using the original area as basis. It is noted that, in principle, damage is a measurable parameter, which could be determined by measuring the damaged area, remaining area, or more practically measuring the initial and remaining moduli. Therefore, in thermodynamic terms, damage is a measurable state variable, in the same sense as the temperature is a measurable state variable that quantifies in macroscopic terms the random agitation of atoms, molecules, and other elementary particles. While it is possible, but extremely difficult, to track the agitation of atoms and molecules, it is very easy to measure the temperature with a thermometer or other device. The same holds true for damage in composite materials.

The analysis of a structural component is done in terms of the nominal area $\widetilde{A}$, which is the only one known to the designer. The remaining area $A = (1-D)\widetilde{A}$ is not known a priori. The nominal stress is $\sigma = P/\widetilde{A}$. Neglecting stress concentrations at the tips of the fictitious cracks representing damage in the damaged configuration (Figure 8.1.b), the value of effective stress[2] acting on the remaining area A is $\widetilde{\sigma} = P/A > P/\widetilde{A}$.

[1] See also (8.10).

[2] Even taking into account stress concentrations, the volume average of the distribution of *effective stress* in the representative volume element (RVE, see Chapter 6) is still $\widetilde{\sigma} = P/A$.

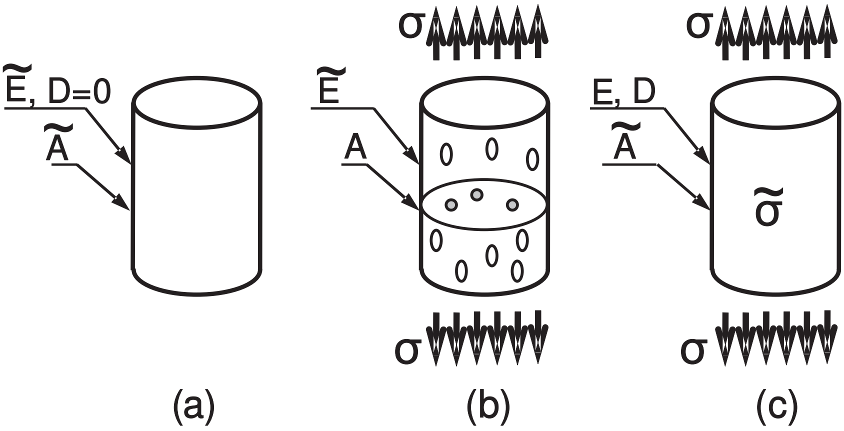


Fig. 8.1: (a) Unstressed material configuration, (b) stressed material configuration with distributed damage, (c) effective configuration.

Therefore, we can envision a configuration (Figure 8.1.c) free of damage, with nominal area $\widetilde{A}$, loaded by nominal stress σ, but internally subjected to effective stress $\widetilde{\sigma}$ and degraded stiffness E. Thus, the effective configuration allows us to perform structural analysis using the nominal geometry but effectively taking into account the increase of effective stress and the decrement of stiffness caused by damage.

In the undamaged configuration (a), $D = 0$, $\sigma = \widetilde{\sigma}$, $\varepsilon = \widetilde{\varepsilon}$, and Hooke's law is

$$\widetilde{\sigma} = \widetilde{E}\,\widetilde{\varepsilon} \tag{8.4}$$

where $\widetilde{E}$ is a constant. In the effective configuration (c)

$$\sigma = E\,\varepsilon \tag{8.5}$$

where E is a function of D. The *principle of strain equivalence* assumes that the strain is the same in the configurations (b) and (c), or $\varepsilon = \widetilde{\varepsilon}$. Starting with the nominal stress $\sigma = P/\widetilde{A}$, multiplying by A/A and using (8.3), the relationship between effective stress $\widetilde{\sigma}$ and nominal stress σ (under strain equivalence) is

$$\sigma = \widetilde{\sigma}\ [1 - D] \tag{8.6}$$

Using (8.6), (8.4), and $\varepsilon = \widetilde{\varepsilon}$ in (8.5), the apparent modulus is a function of the damage D

$$E(D) = \widetilde{E}\ [1 - D] \tag{8.7}$$

The *principle of energy equivalence* [59] states that the elastic strain energy is identical in the configurations (b) and (c). That is, $\sigma : \varepsilon = \widetilde{\sigma} : \widetilde{\varepsilon}$, which is satisfied by

$$\begin{aligned} \sigma &= \widetilde{\sigma}\ [1 - D] \\ \widetilde{\varepsilon} &= \varepsilon\ [1 - D] \end{aligned} \tag{8.8}$$

Substituting (8.8) in (8.5) yields

$$E(D) = \widetilde{E}\ [1 - D]^2 \tag{8.9}$$

which redefines the damage variable as

$$D = 1 - \sqrt{E/\widetilde{E}} \tag{8.10}$$

Every state variable has a conjugate thermodynamic force driving its growth. In plasticity, the measurable state variable is the plastic strain tensor ε^p, which is driven to grow by its conjugate thermodynamic force, the stress tensor σ. The thermodynamic damage force Y is defined as conjugate to the state variable D.

A kinetic equation $\dot{D}(Y)$ governs the growth of the state variable D as a function of its conjugate thermodynamic force Y. In principle, any relevant variable can be chosen as independent variable Y to define the kinetic equation $\dot{D}(Y)$, as long as it is independent of its conjugate state variable. When the damage D is a scalar and it is used to analyze one-dimensional problems, various authors have chosen independent variables in the form of strain ε [60], effective stress $\widetilde{\sigma}$ [61, 62], excess energy release rate $G - 2\gamma_c$ [63], and so on. However, the choice is better based on the appropriate form of the thermodynamic principle governing the problem, as it is shown in Section 8.3.

8.1.2 Damage Threshold and Activation Function

The elastic domain is defined by a threshold value for the thermodynamic force below which no damage occurs. When the load state is in the elastic domain, damage does not grow. When the load state reaches the limit of the elastic domain, additional damage occurs. Furthermore, the elastic domain modifies its size or hardens. Typical one-dimensional responses of two materials are shown in Figure 8.2. Initially the elastic domain is defined by the initial threshold values, $\sigma \leq \sigma_0$ and $\varepsilon \leq \varepsilon_0$. While the load state is inside this domain, no damage occurs. When the load state is higher than the threshold, damage increases and the threshold changes. The elastic domain may evolve as hardening or softening. A stress threshold increases for materials with hardening (see Figure 8.2a), and it decreases for materials with softening (see Figure 8.2b). On the other hand, a strain or effective stress threshold always increases for hardening and softening behavior, as shown in Figure 8.2.

The elastic domain can be defined by the *damage activation function* g as

$$g = \hat{g} - \hat{\gamma} \leq 0 \tag{8.11}$$

where $\hat{g}$ is a positive function (norm) that depends on the independent variable (in a one-dimensional case a scalar Y) and $\hat{\gamma}$ is the updated damage threshold for isotropic hardening. According to the positive dissipation principle (see Section 8.3 and (8.82),(8.97)), the updated damage threshold $\hat{\gamma}$ can be written as

$$\hat{\gamma} = \gamma(\delta) + \gamma_0 \tag{8.12}$$

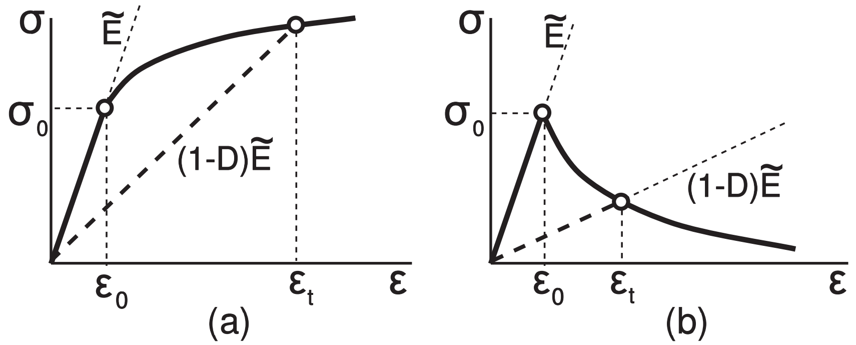


Fig. 8.2: (a) Hardening behavior and (b) softening behavior. No damage occurs until the strain reaches a threshold value ε_0, and no damage occurs during unloading.

where γ_0 denotes the virgin damage threshold, and γ is a positive monotonic function, called *hardening (or softening) function*, that depends on the internal variable δ, called *damage hardening variable.*

8.1.3 Kinetic Equation

The rate of damage accumulation is represented by a kinetic equation. The evolution of damage and hardening are defined by

$$\dot{D} = \dot{\lambda}\frac{\partial g}{\partial Y} \quad ; \qquad \dot{\delta} = \dot{\lambda}\frac{\partial g}{\partial \gamma} \tag{8.13}$$

where Y is the independent variable and $\dot{\lambda} \geq 0$ is the damage multiplier that enforces consistency among the damage and hardening evolution as defined by (8.13). Furthermore, the values of $\dot{\lambda}$ and g allow us to distinguish among two possible situations, loading or unloading without damage growth, and loading with damage growth, according to the Kuhn-Tucker conditions [64]

$$\dot{\lambda} \geq 0 \quad ; \quad g \leq 0 \quad ; \quad \dot{\lambda} g = 0 \tag{8.14}$$

In other words, the Kuhn-Tucker conditions allow us to differentiate among two different cases:

i. Un-damaging loading or unloading, in the elastic domain. The damage activation function is $g < 0$ and by condition (8.14.c) $\dot{\lambda} = 0$, and by (8.13.a) $\dot{D} = 0$.

ii. Damage loading. In this case $\dot{\lambda} > 0$ and condition (8.14.c) implies that $g = 0$. Then, the value of $\dot{\lambda}$ can be determined by the damage consistency condition

$$g = 0 \quad \text{and} \quad \dot{g} = 0 \tag{8.15}$$

Example 8.1 *Compute $\dot{\lambda}$ for a one-dimensional model under tensile load where the independent variable is the effective stress $Y = \widetilde{\sigma}$, the activation function is defined by $\hat{g} = \widetilde{\sigma}$, and the hardening function is defined by*

$$\hat{\gamma} = (F_0 - F_R)\delta + F_0$$

where F_0 and F_R are the initial threshold and the strength of the strongest microscopic element in the material, respectively.

Solution to Example 8.1 *The damage activation function g is defined as*

$$g = \hat{g} - \hat{\gamma} = \widetilde{\sigma} - [(F_0 - F_R)\delta + F_0] \leq 0$$

Therefore,

$$\frac{\partial g}{\partial \widetilde{\sigma}} = +1 \quad ; \qquad \frac{\partial g}{\partial \hat{\gamma}} = -1$$

Using (8.13), the kinetic equations can be written as

$$\dot{D} = \dot{\lambda}\frac{\partial g}{\partial \widetilde{\sigma}} = \dot{\lambda} \quad ; \qquad \dot{\delta} = \dot{\lambda}\frac{\partial g}{\partial \hat{\gamma}} = -\dot{\lambda}$$

When new damage appears, the consistency conditions (8.15) yield

$$g = 0 \quad \Rightarrow \quad \hat{\gamma} = \widetilde{\sigma}$$

and

$$\dot{g} = 0 \quad \Rightarrow \quad \dot{g} = \frac{\partial g}{\partial \widetilde{\sigma}}\dot{\widetilde{\sigma}} + \frac{\partial g}{\partial \hat{\gamma}}\dot{\hat{\gamma}} = \dot{\widetilde{\sigma}} - \dot{\hat{\gamma}} = 0$$

where

$$\dot{\hat{\gamma}} = \frac{\partial \hat{\gamma}}{\partial \delta}\dot{\delta} = (F_0 - F_R)(-\dot{\lambda}) = (F_R - F_0)\dot{\lambda}$$

Substituting into the second consistency condition (8.15) we obtain $\dot{\lambda}$ as

$$\dot{\lambda} = \frac{1}{F_R - F_0}\dot{\widetilde{\sigma}}$$

8.1.4 Statistical Interpretation of the Kinetic Equation

Let's assume that individual damaging events are caused by the failure of microscopic elements inside the material (e.g., fiber breaks, matrix cracks, fiber-matrix de-bond, etc.). Furthermore, assume each of these material points has a failure strength $\widetilde{\sigma}$ and that the collection of failure strengths for all these points, i.e., elements failing at a certain stress $\widetilde{\sigma}$ over the total number of elements available, is represented by a probability density $f(\widetilde{\sigma})$ (Figure 8.3.b). The fraction of elements broken during an effective stress excursion from zero to $\widetilde{\sigma}$ provides a measure of damage

$$D(\widetilde{\sigma}) = \int_0^{\widetilde{\sigma}} f(\sigma')d\sigma' = F(\widetilde{\sigma}) \tag{8.16}$$

where $F(\widetilde{\sigma})$ is the cumulative probability (Figure 8.3.b) corresponding to the probability density $f(\widetilde{\sigma})$, and σ' is a dummy integration variable. Then, the kinetic equation in terms of effective stress $\widetilde{\sigma}$ is

$$\dot{D} = \frac{dD}{d\widetilde{\sigma}}\dot{\widetilde{\sigma}} = f(\widetilde{\sigma})\dot{\widetilde{\sigma}} \tag{8.17}$$

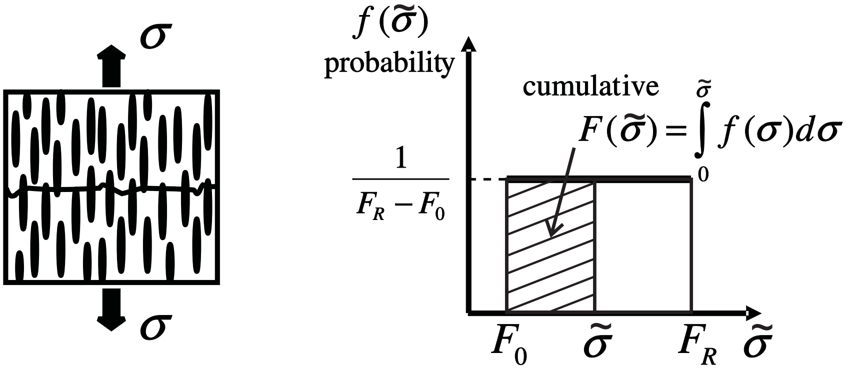


Fig. 8.3: One-dimensional random-strength model.

8.1.5 One-Dimensional Random-Strength Model

As explained in Section 8.1.3, the rate of damage accumulation is represented by a kinetic equation. Equation (8.17) represents a generic kinetic equation, which becomes specific once a particular probability density of failure is adopted.

Consider a loose bundle of short fibers embedded in a matrix and subjected to a uniform stress. The fiber-matrix interfacial strength is assumed to be identical for all fibers but the embedment length is random. The fiber pull out strength is therefore random. Random means that the probability of finding a fiber pulling out at any value of stress $F_0 < \widetilde{\sigma} < F_R$ is constant. In other words, there is no stress level at which more fibers or less fibers pull out because the probability of pull out is random. This is represented in Figure 8.3 and given by the equation $f(\widetilde{\sigma}) = 1/(F_R - F_0)$. Substituting $\widetilde{\varepsilon}$ for $\widetilde{\sigma}$ as the independent variable in (8.17), and assuming strain equivalence $\varepsilon = \widetilde{\varepsilon}$, we have

$$f(\widetilde{\varepsilon}) = \frac{\widetilde{E}}{F_R - F_0} \quad ; \quad F_0 \leq \widetilde{\sigma} \leq F_R \tag{8.18}$$

Equation (8.18) yields the model proposed in [60], which represents well the damaging behavior of Haversian bone [65], concrete in tension [66], fiber composites when damage is controlled by fiber pull out [67], and transverse damage of unidirectional composites.

Damage Activation Function

For a one-dimensional problem, choosing strain as the independent variable, it is possible to write $\hat{g} = \varepsilon$. Therefore, the damage activation function can be written as

$$g = \varepsilon - \hat{\gamma} \leq 0 \tag{8.19}$$

where $\hat{\gamma}$ is the updated damage threshold. Assuming that no damage occurs until the strain reaches a threshold value $\varepsilon_0 = F_0/\widetilde{E}$, and applying the consistency conditions (8.15) and using (8.19), the updated damage threshold $\hat{\gamma}$ is given by the highest value of strain seen by the material, or

$$\hat{\gamma} = \max(\varepsilon_0, \varepsilon) \tag{8.20}$$

Kinetic Equation

The kinetic equation (8.17) for the case of random strength (8.18) in terms of strains $\widetilde{\varepsilon} = \varepsilon$ can be expressed as

$$\dot{D} = \frac{dD}{d\varepsilon}\dot{\varepsilon} = \begin{cases} \widetilde{E}/(F_R - F_0)\dot{\varepsilon} & \text{when ; } \varepsilon > \hat{\gamma} \\ 0 & \text{otherwise} \end{cases} \tag{8.21}$$

In this case, the independent variable is ε, and using (8.19), the kinetic equation (8.13) reduces to

$$\dot{D} = \dot{\lambda} \tag{8.22}$$

Using the Kuhn-Tucker conditions and (8.21), the consistency condition (8.15) reduces to

$$\dot{\lambda} = \widetilde{E}/(F_R - F_0)\dot{\varepsilon} \tag{8.23}$$

when damage occurs and $\dot{\lambda} = 0$ otherwise. In this particular case, the kinetic equation is known explicitly (8.22-8.23). Therefore, it is not necessary to evaluate the evolution of hardening (8.12) because hardening is computed explicitly by (8.20). Note that (8.23) is identical to the solution of Example 8.1 because the hardening function was chosen deliberately to yield this result.

Secant Constitutive Equation

In this particular case, the damage variable is active when tensile load appears, and it can be obtained by integrating (8.21) as

$$D_t = \widetilde{E}\,\frac{\hat{\gamma} - \varepsilon_0}{F_R - F_0} \quad \text{when } \varepsilon > 0 \tag{8.24}$$

Note that the damage state does not depend on the actual load state ε; it only depends on the history of the load state $\hat{\gamma}$. In this example, crack closure is assumed in compression, damage becomes passive, and $D_c = 0$. Mathematically, damage under unilateral contact conditions can be defined by the following equation

$$D = D_t\frac{\langle\varepsilon\rangle}{|\varepsilon|} + D_c\frac{\langle-\varepsilon\rangle}{|\varepsilon|} \tag{8.25}$$

where the McCauley operator $\langle x\rangle$ is defined as $\langle x\rangle := \frac{1}{2}(x + |x|)$.

Substituting (8.24) into (8.5), and using strain equivalence, yields the following constitutive equation

$$\sigma = E(D)\,\varepsilon = \begin{cases} \left(1 - \widetilde{E}\,\frac{\hat{\gamma} - \varepsilon_0}{F_R - F_0}\right)\widetilde{E}\,\varepsilon & \text{when } \varepsilon > 0 \\ \widetilde{E}\,\varepsilon & \text{when } \varepsilon < 0 \end{cases} \tag{8.26}$$

Tangent Constitutive Equation

In a finite element formulation, it is necessary to provide the constitutive equation in rate form, where the rates of stress $\dot{\sigma}$ and strain $\dot{\varepsilon}$ are expressed as functions of pseudotime. In this particular case, the tangent constitutive equation can be obtained by differentiation of the secant constitutive equation as

$$\dot{\sigma} = E(D)\dot{\varepsilon} + \dot{E}(D)\varepsilon \tag{8.27}$$

The term $\dot{E}(D)$ is zero when new damage does not appear, i.e., when there is elastic loading or unloading. When damaging behavior occurs (8.20) yields $\hat{\gamma} = \varepsilon$, and differentiating $E(D)$ in (8.26) we obtain

$$\dot{E}(D) = -\frac{\widetilde{E}^2}{F_R - F_0}\dot{\varepsilon} \tag{8.28}$$

Substituting (8.28) into (8.27) if damage occurs, or $\dot{E}(D) = 0$ if no damage occurs, the tangent constitutive equation can be written as

$$\dot{\sigma} = \begin{cases} \left(1 - \widetilde{E}\,\dfrac{2\hat{\gamma} - \varepsilon_0}{F_R - F_0}\right) \widetilde{E}\,\dot{\varepsilon} & \text{when } \varepsilon > \hat{\gamma} \\ E(D)\,\dot{\varepsilon} & \text{when } \varepsilon < \hat{\gamma} \end{cases} \tag{8.29}$$

Model Identification

The initial damage threshold ε_0 represents the minimum strain to initiate damage and it is proportional to F_0 as follows

$$F_0 = \widetilde{E}\varepsilon_0 \tag{8.30}$$

Under load control, a tensile specimen breaks at $\varepsilon = \hat{\gamma} = \varepsilon_{cr}$ when $d\sigma/d\varepsilon = 0$. Then, using (8.29.a), the only unknown parameter in the model can be computed as

$$F_R = 2\widetilde{E}\varepsilon_{cr} \tag{8.31}$$

The material parameters F_0 and F_R can be calculated from the experimental data using (8.30) and (8.31) with $\widetilde{E}$ being the undamaged modulus of the material. The measurable values ε_0 and ε_{cr} can be obtained easily from material testing at the macroscopic level.

For the particular case $\varepsilon_0 = 0$, using (8.24) and (8.31) at $\varepsilon = \varepsilon_{cr}$, the critical damage at failure under tensile load is

$$D_{cr} = 0.5 \tag{8.32}$$

Therefore the critical effective stress is

$$\widetilde{\sigma}_{T\ cr} = \widetilde{E}\varepsilon_{cr} = 0.5F_R \tag{8.33}$$

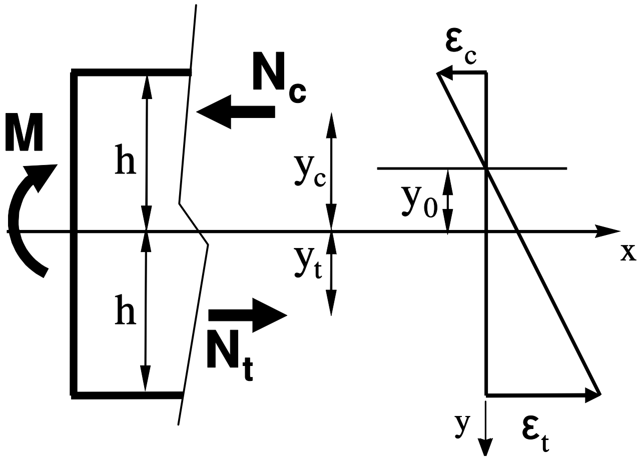


Fig. 8.4: One-dimensional random-strength model.[3]

and using (8.7) the critical applied stress is

$$\sigma_{T\ cr} = 0.25 F_R \tag{8.34}$$

Therefore, in a material with initial threshold $\varepsilon_0 = 0$, a tensile specimen under load control fails when $D = 1/2$ and applied stress $F_R/4$.

A conservative estimate of transverse tensile strength of a fiber reinforced lamina can be obtained assuming that the fiber-matrix bond strength is negligible. In the limit, only the matrix between fibers carries the transverse load, with the fibers acting as holes. In this limit case, the matrix links can be assumed to have a random distribution of strength (8.18). Therefore, the random-strength model (8.29) applies, and the critical damage for transverse tensile loading of unidirectional fiber-reinforced lamina can be estimated by (8.32) as $D_{2t}^{cr} = 0.5$. At the present time, there is no model available to estimate the critical transverse-direction compression damage D_{2c}^{cr}.

Example 8.2 *A beam of rectangular cross section, width* $b = 100$ *mm and height* $2h = 200$ *mm, is subjected to pure bending. The bending moment at failure is* 25.1 *MN mm. The beam is made of carbon/epoxy composite with randomly oriented short fibers with undamaged Young's modulus* $\widetilde{E} = 46$ *GPa. Find the bending moment at failure in terms of* F_R *in (8.21). Assume that the material does not damage in compression and it has a random distribution of strength in tension, with the strongest material element having unknown strength* $F_R > 0$ *and* $F_0 = 0$*. Determine* F_R *using the data given.*

Solution to Example 8.2 *This problem was solved in [67]. With reference to Figure 8.4,* M *is the applied bending moment, and* y_c*,* y_t *are the distances from the neutral axis to the stress resultants* N_c*,* N_t *on the tensile and compression portions of the beam.*

Denoting by ε_t *and* ε_c *the tension and compression strain on the outer surfaces of the beam,* y_0 *the distance from the mid-plane to the neutral surface, and assuming linear strain*

[3] Reprinted from *Mechanics of Materials*, vol. 8, issue 2-3, D. Kracjcinovic, Damage Mechanics, Fig. 2.11, p. 134, copyright (1989), with permission from Elsevier.

distribution through the thickness we have

$$\varepsilon(y) = \frac{y - y_0}{h - y_0}\varepsilon_t \quad or \quad \varepsilon(y) = \frac{-y + y_0}{h + y_0}\varepsilon_c$$

Since there is no damage in compression, the compression stress distribution is linear, and the resulting compression stress resultant is

$$N_c = \frac{1}{2}b(h + y_0)\widetilde{E}\varepsilon_c$$

and the distance y_c *is*

$$y_c = \frac{1}{3}(y_0 - 2h)$$

As the tensile side of the beam damages, the neutral axis moves away from the mid-surface. The tensile stress resultant is obtained using (8.26) and integrating the stress between y_0 *and* h *as*

$$N_t = \int_{y_0}^{h} dN_t = b\int_{y_0}^{h} E(D)\varepsilon(y)dy = \frac{1}{6}(h - y_0)b\left(3 - 2(\widetilde{E}/F_R)\varepsilon_t\right)\widetilde{E}\varepsilon_t$$

where $\widetilde{E}$ *is the undamaged elastic moduli. The distance* y_t *is*

$$y_t = \frac{1}{N_t}\int_{y_0}^{h} y dN_t = \frac{4h - 2y_0 - (\widetilde{E}/F_R)\varepsilon_t\,(3h + y_0)}{6 - 4(\widetilde{E}/F_R)\varepsilon_t}$$

The force and moment equations of equilibrium are

$$N_c + N_t = 0$$
$$N_c y_c + N_t y_t = M$$

Using the force equilibrium equation and assuming linear strain distribution through the thickness, it is possible to obtain the strains ε_t *and* ε_c *in terms of* y_0 *as*

$$\varepsilon_t = -\frac{6hy_0}{(h - y_0)^2}\frac{F_R}{\widetilde{E}} \quad ; \quad \varepsilon_c = \frac{6hy_0(h + y_0)}{(h - y_0)^3}\frac{F_R}{\widetilde{E}}$$

Using the above relation, it is possible to reduce the moment equilibrium equation to a single cubic equation in y_0

$$M = \frac{-y_0(4h^2 + 9hy_0 + 3y_0^2)}{(h - y_0)^3}\,bh^2F_R$$

The ultimate bending moment can be determined by differentiation with respect to y_0

$$\frac{dM}{dy_0} = 0$$

that yields $y_{0\,cr} = -0.175\,h$ *at beam failure. Therefore, the rupture bending moment is*

$$M_{cr} = 0.2715\,bh^2\,F_R$$

A simple test (ASTM D790 or D6272) can be used to obtain the bending moment at failure; in this example $M_{cr} = 25.1\;10^6 N\,mm$. *Therefore,* F_R *can be estimated as* $F_R = 92\;MPa$. *As it is customary in structural engineering, the equivalent bending strength is defined as*

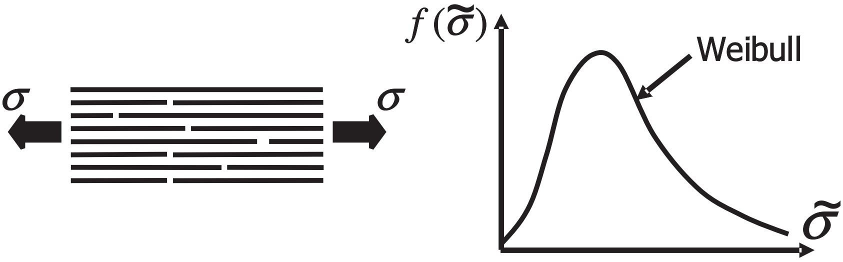


Fig. 8.5: One-dimensional random-strength model.

$$\sigma_{Bcr} = \frac{M_{cr}}{S} = 0.407\ F_R$$

where S is the section modulus (for a rectangular beam $S = \frac{2}{3}bh^2$). Note that according to (8.34), the tensile strength of the same material assuming the same kinetic equation (8.26) would be $\sigma_{T\ cr} = 0.25\ F_R$. This gives a ratio of equivalent bending to tensile strength equal to $\sigma_{B\ cr}/\sigma_{T\ cr} = 1.63$, which is in good agreement with experimental data $\sigma_{B\ cr}/\sigma_{T\ cr} = 1.6$ [68] obtained for unreinforced concrete and also with the value $\sigma_{B\ cr}/\sigma_{T\ cr} = 1.5$ recommended by the ACI Code [69].

8.1.6 Fiber-Direction, Tension Damage

If a lamina is subjected to tensile stress in the fiber direction, it is reasonable to assume that the matrix carries only a small portion of the applied load and no damage is expected in the matrix during loading. Then, the ultimate tensile strength of the composite lamina can be accurately predicted by computing the strength of a bundle of fibers.

Fiber strength is a function of the gauge length used during fiber strength tests. The length scale that determines how much of the fiber strength is actually used in a composite is the ineffective length δ. Starting at a fiber break point, the ineffective length is that length over which a fiber recovers a large percentage of its load (say, 90%). Rosen [70] recognized this fact and proposed that the longitudinal ultimate strength of fibers embedded in a ductile-matrix can be accurately predicted by the strength of a dry bundle of fibers with length δ.

A dry bundle is defined as a number of parallel fibers of some given length and diameter which, if unbroken, carry the same load. After a fiber within a dry bundle fails, the load is shared equally by the remaining unbroken fibers. A dry bundle typically refers to fibers which have not yet been combined with a matrix. As tensile load is slowly applied to a dry bundle of fibers, the weaker fibers (with large flaw sizes) begin to fail and the stress on the remaining unbroken fibers increases accordingly. The Weibull expression [71]

$$F(\widetilde{\sigma}) = 1 - \exp\left(-\frac{\delta}{L_0}\left(\frac{\widetilde{\sigma}}{\widetilde{\sigma}_0}\right)^m\right) \tag{8.35}$$

is often used to describe the cumulative probability $F(\widetilde{\sigma})$ that a fiber of length δ will fail at or below an effective stress $\widetilde{\sigma}$. The values of $\widetilde{\sigma}_0$ and m, which represent the characteristic strength of the fiber, and the dispersion of fiber strength, respectively, can be determined from fiber strength experiments performed with a gauge length L_0. Equation (8.35) can be simplified as

$$F(\widetilde{\sigma}) = 1 - \exp\left(-\delta\alpha\widetilde{\sigma}^m\right) \tag{8.36}$$

where

$$\alpha = \frac{1}{L_0\widetilde{\sigma}_0^m} = \left[\frac{\Gamma(1+1/m)}{\widetilde{\sigma}_{av}}\right]^m \frac{1}{L_0} \tag{8.37}$$

where $\Gamma(x)$ is the Gamma function [46]; $\widetilde{\sigma}_{av}$ is the average strength for a gauge length L_0. Equation (8.36) provides the percentage of fibers in a bundle which are broken as a function of the actual (or apparent) stress in the unbroken fibers. The percentage of fibers which are unbroken is $1 - F(\widetilde{\sigma})$. The apparent stress or bundle stress $\sigma = \sigma_b$ is equal to the applied load divided by the total fiber cross-sectional area. It is also equal to the product of the stress in unbroken fibers and the percentage of fibers which are unbroken

$$\sigma = \sigma_b = \widetilde{\sigma}\exp(-\delta\alpha\widetilde{\sigma}^m) \tag{8.38}$$

The value $\widetilde{\sigma}_{\max}$, which maximizes (8.38), can be easily determined and is given by

$$\widetilde{\sigma}_{\max} = (\delta\alpha m)^{-1/m} \tag{8.39}$$

The maximum (or critical) bundle strength σ_{cr} is determined by substituting (8.39) into (8.38)

$$\sigma_{cr} = (\delta\alpha m)^{-1/m}\exp(-1/m) = (\delta\alpha m e)^{-1/m} \tag{8.40}$$

where e is the basis of the natural logarithms. The composite longitudinal tensile strength is [1, (4.82)]

$$F_{1t} = \left[V_f + \frac{E_m}{E_f}(1 - V_f)\right]\sigma_{cr} \tag{8.41}$$

where V_f is the fiber volume fraction, and E_f and E_m are the fiber and matrix elastic Young's moduli, respectively.

Combining (8.36) and (8.39), we get

$$D_{1t}^{cr} = 1 - \exp(-1/m) \tag{8.42}$$

Therefore, the critical or maximum damage D_{1t}^{cr} for longitudinal tensile loading can be computed as the area fraction of broken fibers in the lamina prior to catastrophic failure [61, 62], which turns out to be a function of the Weibull shape modulus m only.

Example 8.3 *The data sheet of carbon fiber T300 from Toray^{TM} Carbon Fibers, Inc. gives average tensile strength of the fiber $\sigma_{av} = 3.53\ GPa$, and tensile modulus $E_f = 230\ GPa$. Also, the same data sheet provides results of tensile tests of a uni-directional (UD) composite with epoxy $E_m = 4.5\ GPa$ and fiber volume fraction $V_f = 0.6$. The tensile strength reported is $F_{1t} = 1860\ MPa$. Using this experimental data, and assuming a Weibull shape parameter $m = 8.9$, identify the damage model under tensile load. Then, formulate the damage model and implement it in* Abaqus *for a one-dimensional bar element. Finally, obtain the strain vs. stress response of the UD composite.*

Solution to Example 8.3

MODEL IDENTIFICATION *From (8.41) and using the experimental data available, it is possible to obtain σ_{cr} as*

$$\sigma_{cr} = \frac{F_{1t}}{V_f + \frac{E_m}{E_f}(1 - V_f)} = 3060\ MPa$$

Then, the product $\delta\alpha$ can be obtained using (8.40) as

$$\delta\alpha = \frac{(\sigma_{cr})^{-m}}{me} = 3.92\ 10^{-33}$$

The properties $E_f = 230\ GPa$, $m = 8.9$, and $\delta\alpha = 3.92 \times 10^{-33}$ are sufficient for the identification of the model.

MODEL FORMULATION *Following a procedure similar to that shown in Section 8.1.5 to implement a damage model, the following items are needed*

Damage Activation Function *In this example, the effective stress is chosen as the independent variable. Therefore, the damage activation function can be written as*

$$g = \widetilde{\sigma} - \hat{\gamma} \leq 0 \tag{8.43}$$

where $\hat{\gamma}$ is the updated damage threshold. Assuming an initial threshold value $\sigma_0 = 0$, from the consistency conditions (8.15) and (8.19), $\hat{\gamma}$ is given by the highest value of effective stress seen by the material

$$\hat{\gamma} = \max(0, \widetilde{\sigma}) \tag{8.44}$$

Secant Constitutive Equation *In this example, the kinetic equation (8.1.3) is available in integral form and given explicitly by (8.36) as*

$$D = 1 - \exp\left(-\delta\alpha\hat{\gamma}^m\right) \quad when\ \widetilde{\sigma} > 0;\ \varepsilon > 0 \tag{8.45}$$

where the damage state does not depend on the actual load state $\widetilde{\sigma}$; it only depends on the load history state $\hat{\gamma}$.

Substituting (8.45) into (8.5) and (8.7), and using strain equivalence, yields the constitutive equation

$$\sigma = E(D)\,\varepsilon = \exp\left(-\delta\alpha\hat{\gamma}^m\right)\,\widetilde{E}\,\varepsilon \quad when \quad \varepsilon > 0 \tag{8.46}$$

Tangent Constitutive Equation *The tangent constitutive equation can be obtained by differentiating the secant constitutive equation as*

$$\dot{\sigma} = E(D)\dot{\varepsilon} + \dot{E}(D)\varepsilon \tag{8.47}$$

The factor $\dot{E}(D)$ *is zero when no new damage appears, i.e., during elastic loading or unloading. When damage occurs, (8.44) yields* $\hat{\gamma} = \widetilde{E}\varepsilon$*, and differentiating* $E(D)$ *in (8.46) we obtain*

$$\dot{E}(D) = -\delta\alpha m\hat{\gamma}^{m-1}\exp\left(-\delta\alpha\hat{\gamma}^{m}\right)\widetilde{E}^{2}\dot{\varepsilon} \tag{8.48}$$

The tangent constitutive equation is obtained by substituting (8.48) into (8.47) when damage occurs, or $\dot{E}(D) = 0$ *when no new damage appears. Therefore, the tangent constitutive equation can be written as*

$$\dot{\sigma} = \begin{cases} (1-\delta\alpha m\hat{\gamma}^{m})\exp\left(-\delta\alpha\hat{\gamma}^{m}\right)\widetilde{E}\,\dot{\varepsilon} & when \quad \varepsilon > \hat{\gamma}/\widetilde{E} \\ E(D)\,\dot{\varepsilon} & when \quad \varepsilon < \hat{\gamma}/\widetilde{E} \end{cases} \tag{8.49}$$

NUMERICAL ALGORITHM *The one-dimensional damage model is implemented in* Abaqus *using the UMAT subroutine* `umat1d83.for`*, available in [5]. The following items describe the procedure used to explicitly evaluate the damage constitutive equation.*[4]

i. Read the strain at time t

$$\varepsilon_t$$

ii. Compute the effective stress (assuming strain equivalence)

$$\widetilde{\sigma}_t = \widetilde{E}\varepsilon_t$$

iii. Update the threshold value

$$\hat{\gamma}_t = \max(\hat{\gamma}_{t-1}, \widetilde{\sigma}_t)$$

iv. Compute the damage variable

$$D_t = 1 - \exp\left(-\delta\alpha(\hat{\gamma}_t)^{m}\right)$$

v. Compute the nominal stress

$$\sigma_t = (1-D_t)\,\widetilde{E}\,\varepsilon_t$$

vi. Compute the tangent stiffness

$$E_t^{dam} = \begin{cases} (1-\delta\alpha m(\hat{\gamma}_t)^{m})\exp\left(-\delta\alpha(\hat{\gamma}_t)^{m}\right)\widetilde{E} & when\ \hat{\gamma}_t > \hat{\gamma}_{t-1} \\ (1-D_t)\,\widetilde{E} & when\ \hat{\gamma}_t = \hat{\gamma}_{t-1} \end{cases}$$

MODEL RESPONSE *The pseudo code below and user subroutine* ***`umat1d83.for`****, available in [5], are used to model a one-dimensional bar representative of a carbon fiber UD composite. The nominal stress-strain response is shown with a solid line in Figure 8.6. The UD composite fails at* $\varepsilon_{cr} = 1.5\%$*, in good agreement with the strain to failure reported by Toray.*

[4] See Section 8.4.1 for those cases for which it is not possible to integrate the constitutive equation explicitly.

i. *Retrieve* `Ex_2.4.cae` *or use the instructions given in Example 2.6 (p. 79) to recreate the model*

```
Menu:   File, Open, [C:\SIMULIA\User\Ex_2.4\Ex_2.4.cae], OK
Menu:   File, Save As, [C:\SIMULIA\User\Ex_8.3\Ex_8.3.cae], OK
Menu:   File, Set Work Directory, [C:\SIMULIA\User\Ex_8.3], OK
```

ii. *Modify the geometry*

```
Module: Part
    # on the left tree, expand (+) the following items:
    # expand: Model-1,  # expand: Parts (1)
    # expand: Part-1,   # expand: Features, # expand: Wire-1
    # right-click: Sketch, Edit
    # use dimension tool to change the length to [10]
Menu:   Add, Dimension
    # pick: Line, # create the dimension below the line, [10]
    X # to close the command, Done
Menu:   Feature, Regenerate
Module: Assembly
Menu:   Feature, Regenerate
```

iii. *Delete the existing Material-1 and create a new Material-1 with the properties and parameters required by the user material. Modify the cross section.*

```
Module: Property
Menu:   Material, Manager
    # select: Material-1, Delete, Yes
    Create, Name [Material-1], General, User Material
    Mechanical Constants [203.0E3  8.9 3.92E-33]
    # or right click Read from File, File: [props.txt]
    General, Depvar, Number of state variables [2],
    # SDV1: effective stress, see umat1d83.for
    # SDV2: damage, see umat1d83.for
    OK, # close Material Manager pop-up window
Menu:   Section, Edit, Section-1, Cross-sectional area: [1.0], OK
```

iv. *Delete the load and add a specified displacement*

```
Module: Load
Menu:   Load, Delete, Load-1, Yes
Menu:   BC, Create
    Name [BC-2], Step: Step-1, Disp/Rota, Cont
    # pick: free-end node, Done, # checkmark: U1 [0.25], OK
```

v. *Remesh with five elements*

```
Module: Mesh
    # to make the instance "independent", on the left tree
    # expand: Assembly, # expand: Instances
    # right-click: Part-1-1, Make Independent
Menu:   Seed, Instance, Approximate global size [2], Apply, OK
Menu:   Mesh, Element Type
    Element Library: Standard, Geometric Order: Linear
    Family: Trus # verify T2D2 is chosen, OK
```

```
Menu:   Mesh, Instance, Yes
Menu:   View, Assembly Display Options
    Tab: Mesh, # checkmark: Show node labels, OK
    # create a set to track the history of some variables
Menu:   Tools, Set, Create
    Name [Set-1], Type: Element, Cont
    # pick: element between nodes 5 and 6, Done
```

vi. *Modify the step to define the incrementation procedure, using 50 steps. A pseudo-time period of 50 is used but any time period could be used because it is not real time. Also, add solution dependent variables (SDV) to the Output Requests.*

```
Module: Step
Step:   Step-1
    # set up an incremental analysis over 50 units of pseudo time
Menu:   Step, Edit, Step-1
    Tab: Basic, Time Period [50]
    Tab: Incrementation, Type: Fixed, Maximum number of incr. [50]
    Increment size [1], OK
Menu:   Output, Field Output Requests, Edit, F-Output-1
    # expand: State/Field/User/Time, # checkmark: SDV, OK
Menu:   Output, History Output Requests, Edit, H-Output-1
    Domain: Set: Set-1
    # uncheckmark: all
    # expand: Stresses, # expand: S, # checkmark: S11
    # expand: Strains,  # expand: E, # checkmark: E11
    # expand: State/Field/User/Time, # checkmark: SDV, OK
```

vii. *Edit the Job to define the user material subroutine*

```
Module: Job
Menu:   Job, Manager
    # select: Job-1, Edit
    Tab: General, User Subroutine, Select, [umat1d83.for], OK, OK
    Submit, # when Completed, Results
```

viii. *Visualize the results. Failure occurs at Increment: 30, with strain* $E = 1.5 \times 10^{-2}$*, Cauchy stress* $\sigma = 3059.0\ MPa$*, effective stress SDV1=3450 MPa, and SDV2, damage, D=0.1132.*

```
Module: Visualization
Menu:   Plot, Contours, On Deformed Shape
Menu:   Result, Step/Frame
    # select: Increment 30 and verify your results
    # close Step/Frame pop-up window
    # plotting the stress vs. strain results
Menu:   Result, History Output
    # select: E11, Save As [e11], OK
    # select: S11, Save As [s11], OK
    # close History Output pop-up window
Menu:   Tools, XYData, Create
    Source: Operate on XY data, Cont, Operators: Combine
    # select: e11, Add to Expr., # select: s11, Add to Expr.
```

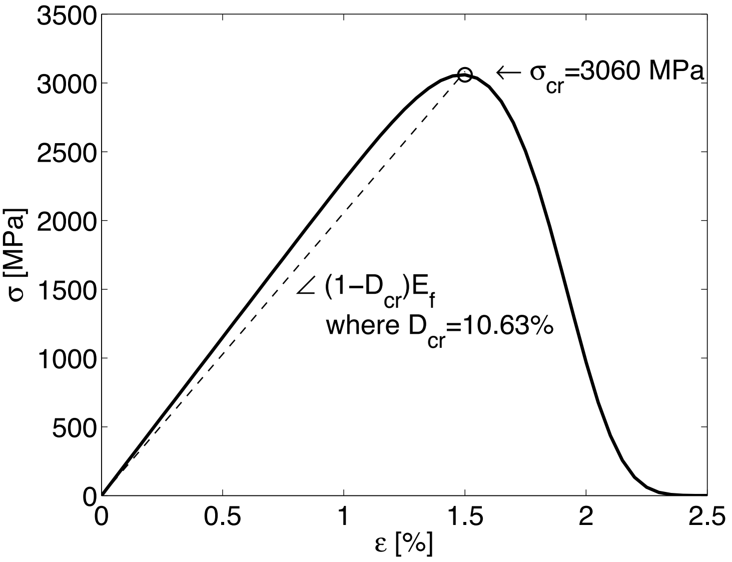


Fig. 8.6: Fiber tensile damage model response.

```
Save As [s11-vs-e11], OK, Plot Expression
```

Your results must be similar to those shown in Figure 8.6.

8.1.7 Fiber-Direction, Compression Damage

Many models have been proposed trying to improve the prediction of compression strength of composites first introduced by Rosen [72]. The literature encompasses fiber buckling modes [21, 25, 73, 74], kink-band models [75], and kink-bands induced by buckling [76]. In fiber buckling models, it is assumed that buckling of the fibers initiates a process that leads to the collapse of the material [72]. Rosen's model has been refined with the addition of initial fiber misalignment and nonlinear shear stiffness [73]. Experimental evidence suggests that fiber buckling of perfectly aligned fibers (Rosen's model) is an imperfection sensitive problem (see Section 4.1.1). Therefore, small amounts of imperfection (misalignment) cause large reductions in the buckling load, thus the reduction of the compression strength with respect to Rosen's prediction. Each fiber has a different value of fiber misalignment. The probability of finding a fiber with misalignment angle α is given by a Gaussian distribution [25, 77].

An optical technique [13] can be used to measure the misalignment angle of each fiber in the cross section. The resulting distribution of fiber misalignment was shown to be Gaussian by using the cumulative distribution function (CDF) plot and the probability plot [25]. Therefore, the probability density is

$$f(z) = \frac{\exp(-z^2)}{\Lambda\sqrt{2\pi}} \quad ; \quad z = \frac{\alpha}{\Lambda\sqrt{2}} \tag{8.50}$$

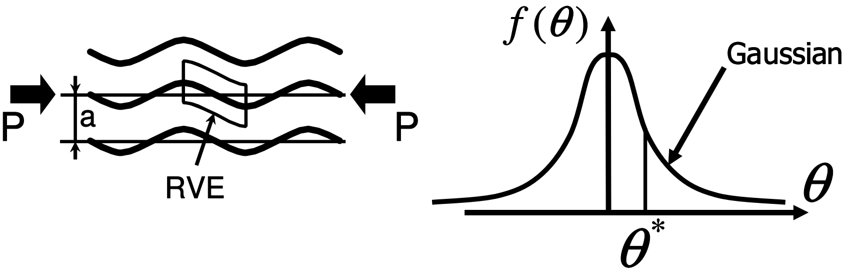


Fig. 8.7: One-dimensional random-strength model.

where Λ is the standard deviation and α is the continuous random variable, in this case being equal to the misalignment angle. The CDF gives the probability of obtaining a value smaller than or equal to some value of α, as follows

$$F(z) = \mathrm{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z exp(-z'^2)dz' \tag{8.51}$$

where $erf(z)$ is the error function.

The relationship between the buckling stress and the imperfection (misalignment) is known in stability theory as the imperfection sensitivity curve. Several models from the literature can be used to develop this type of curve. The deterministic model, similar to the one presented by Wang [73], is developed in [21] but using the representation of the shear response given by Equation (8.52).

The shear stress-strain response of polymer-matrix composites can be represented [25, 76] by

$$\sigma_6 = F_6 \tanh\left(\frac{G_{12}}{F_6}\gamma_6\right) \tag{8.52}$$

where γ_6 is the in-plane shear strain. Furthermore, G_{12} is the initial shear stiffness and F_6 is the shear strength, which are obtained by fitting the stress-strain experimental data. Complete polynomial expansions [78] fit the experimental data well but they are not antisymmetric with respect to the origin. This introduces an artificial asymmetric bifurcation during the stability analysis [74]. Shear experimental data can be obtained by a variety of techniques including the ±45 coupon, 10^o off-axis, rail shear, Iosipescu, Arcan, and torsion tests [79]. The nonlinear shear stress-strain curve should be measured for the actual composite being tested in compression.

Barbero [21] derived the equilibrium stress σ_{eq} as a function of the shear strain and the misalignment angle as

$$\begin{aligned}
\widetilde{\sigma}_{eq}(\alpha,\gamma_6) = & \ \frac{F_6}{2(\gamma_6+\alpha)}\frac{(\sqrt{2}-1)a+(\sqrt{2}+1)(b-1)}{1-a+b} \\
a = & \ \exp(\sqrt{2}g)-\exp(2g) \\
b = & \ \exp(2g+\sqrt{2}g) \\
g = & \ \frac{\gamma_6 G_{12}}{F_6}
\end{aligned} \tag{8.53}$$

with G_{12} and F_6 as parameters. Note that if the shear behavior is assumed to be linear $\widetilde{\sigma}_6 = G_{12}\gamma_6$ [80], then (8.53) does not have a maximum with respect to γ_6 and thus misaligned fibers embedded in a linear elastic matrix do not buckle. On the contrary, by using the hyperbolic tangent representation of shear (8.52), a maximum with respect to γ_6 is shown in (8.53). The maxima of the curves $\widetilde{\sigma}(\gamma_6)$ as a function of the misalignment angle α is the imperfection sensitivity curve, which represents the compression strength of a fiber (and surrounding matrix) as a function of its misalignment. For negative values of misalignment, it suffices to assume that the function is symmetric $\widetilde{\sigma}(-\alpha) = \widetilde{\sigma}(\alpha)$.

The stress carried by a fiber reduces rapidly after reaching its maximum because the load carrying capacity of a buckled fiber is much lower than the applied load. Several models can be constructed depending on the assumed load that a fiber carries after buckling. A lower bound can be found assuming that buckled fibers carry no more load because they have no post-buckling strength. According to the imperfection sensitivity equation (8.53), fibers with large misalignment buckle under low applied stress. If the post-buckling strength is assumed to be zero, the applied stress is redistributed onto the remaining, unbuckled fibers, which then carry a higher effective stress $\widetilde{\sigma}(\alpha)$. At any time during loading of the specimen, the applied load σ (applied stress times initial fiber area) is equal to the effective stress times the area of fibers that remain unbuckled

$$\sigma = \bar{\sigma}(\alpha)[1-D(\alpha)] \tag{8.54}$$

where $0 \le D(\alpha) \le 1$ is the area of the buckled fibers per unit of initial fiber area. For any value of effective stress, all fibers having more than the corresponding value of misalignment have buckled. The area of buckled fibers $D(\alpha)$ is proportional to the area under the normal distribution located beyond the misalignment angle $\pm\alpha$.

Equation (8.54) has a maximum that corresponds to the maximum stress that can be applied to the composite. Therefore, the compression strength of the composite is found as

$$\sigma_c = \max\left[\bar{\sigma}(\alpha)\int_{-\alpha}^{\alpha} f(\alpha')d\alpha'\right] \tag{8.55}$$

where $\bar{\sigma}(\alpha)$ is given by Equation (8.53) and $f(\alpha')$ is given by (8.50). The maximum of (8.54), given by Equation (8.55), is a unique value for the compression strength of the composite that incorporates both the imperfection sensitivity and the distribution of fiber misalignment. Note that the standard deviation Λ is a parameter that describes the actual, measured distribution of fiber misalignment, and it is not to be chosen arbitrarily as a representative value of fiber misalignment for all the fibers.

Since the distribution given in (8.50) cannot be integrated in closed form, (8.55) is evaluated numerically. However, it is advantageous to develop an explicit formula so that the compression strength can be easily predicted. Following the explicit formulation in [21], the compression strength of the unidirectional composite, explicitly in terms of the standard deviation of fiber misalignment Λ, the in-plane shear stiffness G_{12}, and the shear strength F_6 is

$$\frac{F_{1c}}{G_{12}} = (1 + 4.76 B_a)^{-0.69} \tag{8.56}$$

where 4.76 and -0.69 are two constants chosen to fit the numerical solution to the exact problem [21], with the dimensionless group B_a given by

$$B_a = \frac{G_{12}\Lambda}{F_6} \tag{8.57}$$

The misalignment angle of the fibers that buckle just prior to compression failure is given by [21, (23)]

$$\begin{array}{lcl}
\alpha_{cr} & = & a/b \\
a & = & 1019.011 G_{12} C_2^2 \Lambda^3 - 375.3162 C_2^3 \Lambda^4 - 845.7457 G_{12}^2 C_2 \Lambda^2 \\
 & & + g\left(282.1113 G_{12} C_2 \Lambda^2 - 148.1863 G_{12}^2 \Lambda - 132.6943 C_2^2 \Lambda^3\right) \\
b & = & 457.3229 C_2^3 \Lambda^3 - 660.77 G_{12} C_2^2 \Lambda^2 - 22.43143 G_{12}^2 C_2 \Lambda \\
 & & + g\left(161.6881 C_2^2 \Lambda^2 - 138.3753 G_{12} C_2 \Lambda - 61.38939 G_{12}^2\right) \\
g & = & \sqrt{C_2 \Lambda \left(8.0 C_2 \Lambda - 9.424778 G_{12}\right)} \\
C_2 & = & -G_{12}^2 / \left(4 F_6\right)
\end{array} \tag{8.58}$$

Additionally, the shear strain at failure is

$$\gamma_{cr} = -\alpha_{cr} + \sqrt{\alpha_{cr}^2 + \frac{3}{2}\frac{\pi F_6 \alpha_{cr}}{G_{12}}} \tag{8.59}$$

In summary, when a fiber-reinforced lamina is compressed, the predominant damage mode is fiber buckling. However, the buckling load of the fibers is lower than that of the perfect system because of fiber misalignment, so much that a small amount of fiber misalignment could cause a large reduction in the buckling load. For each misalignment angle α, the composite area-fraction with buckled fibers $D(\alpha)$, corresponding to fibers with misalignment angle greater than α, can be taken as a measure of damage. If the fibers are assumed to have no post-buckling strength, then the applied stress is redistributed onto the remaining unbuckled fibers, which will be carrying a higher effective stress. The applied stress, which is lower than the effective stress by the factor $(1 - D)$, has a maximum, which corresponds to the compressive strength of the composite. Therefore, it is possible to compute the critical damage D_{1c} for longitudinal compressive loading as

$$D_{1c}^{cr} = 1 - \Omega_{1c} = 1 - erf\left(\frac{\alpha_{cr}}{\Lambda\sqrt{2}}\right) \tag{8.60}$$

where erf is the error function, Λ is the standard deviation of the actual Gaussian distribution of fiber misalignment (obtained experimentally [77]), and α_{cr} is the critical misalignment angle at failure. The three-dimensional theoretical formulation is developed in the next three sections.

8.2 Multidimensional Damage and Effective Spaces

The first step in the formulation of a general multidimensional damage model is to define the damage variable as well as the effective stress and strain spaces, as shown in this section. The second step is to define the form of either the Helmholtz free energy or the Gibbs energy and from them derive the thermodynamic forces conjugate to the state variables representing damage and hardening, as shown in Section 8.3. The third step is to derive the kinetic laws governing the rate of damage hardening, which are functions of the damage and hardening potentials, as shown in Section 8.4.

Experimental knowledge of the degradation and subsequent material response is used to guide the selection on the variable used to represent damage. A second-order damage tensor **D** can be used to represent damage of orthotropic fiber-reinforced composite materials, following Kachanov-Rabotnov's approach [58, 81]. For composite materials reinforced with stiff and strong fibers, damage can be accurately represented by a second-order tensor[5] with principal directions aligned with the material directions $(1, 2, 3)$ [82–86]. This is due to the fact that the dominant modes of damage are microcracks, fiber breaks, and fiber-matrix debond, all of which can be conceptualized as cracks either parallel or perpendicular to the fiber direction.[6] Therefore, the damage tensor can be written as

$$\mathbf{D} = D_{ij} = D_i \delta_{ij} \quad \text{no sum on i} \tag{8.61}$$

where D_i are the eigenvalues of **D**, which represent the net stiffness reduction along the principal material directions n_i, and δ_{ij} is the Kronecker delta ($\delta_{ij} = 1$ if $i = j$, or zero otherwise). The integrity tensor is also diagonal, and using energy equivalence (8.8), we have

$$\mathbf{\Omega} = \Omega_{ij} = \sqrt{1 - D_i}\delta_{ij} \quad \text{no sum on i} \tag{8.62}$$

The integrity tensor is always symmetric and positive, because the net area reduction must be positive definite during damage evolution [87]. Both tensors are diagonal when represented in the principal system. Introducing a symmetric fourth-order tensor, **M**, called the *damage effect tensor*, as

[5] Tensors are denoted by boldface type, or by their components with index notation.

[6] Strictly speaking, damage is transversely isotropic since cracks can be aligned along any direction in the 2-3 plane.

$$\mathbf{M} = M_{ijkl} = \frac{1}{2}\left(\Omega_{ik}\Omega_{jl} + \Omega_{il}\Omega_{jk}\right) \tag{8.63}$$

The transformation of stress and strain between the effective ($\widetilde{\sigma}$) and damaged (σ) configurations is accomplished as follows

$$\widetilde{\boldsymbol{\sigma}} = \mathbf{M}^{-1} : \boldsymbol{\sigma} \qquad \begin{aligned} \widetilde{\boldsymbol{\varepsilon}} &= \mathbf{M} : \boldsymbol{\varepsilon} \\ \widetilde{\boldsymbol{\varepsilon}}^{e} &= \mathbf{M} : \boldsymbol{\varepsilon}^{e} \\ \widetilde{\boldsymbol{\varepsilon}}^{p} &= \mathbf{M} : \boldsymbol{\varepsilon}^{p} \end{aligned} \tag{8.64}$$

where an over-bar indicates that the quantity is evaluated in the effective configuration and the superscripts e, p, denote quantities in the elastic and plastic domains, respectively.

By the energy equivalence hypothesis [59], it is possible to define the constitutive equation in the effective configuration (Figure 8.1.c) as

$$\widetilde{\boldsymbol{\sigma}} = \widetilde{\mathbf{C}} : \widetilde{\boldsymbol{\varepsilon}}^{e} \quad ; \qquad \widetilde{\boldsymbol{\varepsilon}}^{e} = \widetilde{\mathbf{C}}^{-1} : \widetilde{\boldsymbol{\sigma}} = \widetilde{\mathbf{S}} : \widetilde{\boldsymbol{\sigma}} \tag{8.65}$$

where the fourth-order tensors $\mathbf{C}$ and $\mathbf{S}$ denote the secant stiffness tensor and compliance tensor, respectively. The stress-strain equations in the damaged configuration (Figure 8.1.b) are obtained by substituting (8.65) into (8.64),

$$\begin{aligned} \boldsymbol{\sigma} &= \mathbf{M} : \widetilde{\boldsymbol{\sigma}} = \mathbf{M} : \widetilde{\mathbf{C}} : \widetilde{\boldsymbol{\varepsilon}}^{e}, & \boldsymbol{\varepsilon}^{e} &= \mathbf{M}^{-1} : \widetilde{\boldsymbol{\varepsilon}}^{e} = \mathbf{M}^{-1} : \widetilde{\mathbf{S}} : \widetilde{\boldsymbol{\sigma}}, \\ \boldsymbol{\sigma} &= \mathbf{M} : \widetilde{\mathbf{C}} : \mathbf{M} : \boldsymbol{\varepsilon}^{e}, & \boldsymbol{\varepsilon}^{e} &= \mathbf{M}^{-1} : \widetilde{\mathbf{S}} : \mathbf{M}^{-1} : \boldsymbol{\sigma}, \\ \boldsymbol{\sigma} &= \mathbf{C} : \boldsymbol{\varepsilon}^{e} & \boldsymbol{\varepsilon}^{e} &= \mathbf{S} : \boldsymbol{\sigma} \end{aligned} \tag{8.66}$$

with

$$\mathbf{C} = \mathbf{M} : \widetilde{\mathbf{C}} : \mathbf{M} \qquad \mathbf{S} = \mathbf{M}^{-1} : \widetilde{\mathbf{S}} : \mathbf{M}^{-1} \tag{8.67}$$

The explicit form of these tensors are presented in Appendix B. Given that the tensor $\mathbf{M}$ is symmetric, the secant stiffness and compliance tensors are also symmetric.

8.3 Thermodynamics Formulation

The damage processes considered in this chapter can be described by a series of equilibrium states reached while the system traverses a nonequilibrium path due to the irreversibility of damage and plasticity. In general, the current state of a system (e.g., stress, stiffness, compliance) depends on the current state (e.g., strain) as well as on the history experienced by the system. This is the case for viscoelastic materials discussed in Chapter 7. However, for damaging and elastoplastic materials, the current state can be described in terms of the current strain and the *effects of history* on the material, which in this chapter are characterized by the damage tensor $\mathbf{D}$ and the plastic strain tensor $\boldsymbol{\varepsilon}^{\mathbf{p}}$.

8.3.1 First Law

The *first law* of thermodynamics states that any increment of internal energy of the system is equal to the heat added to the system minus the work done by the system on its surroundings

$$\delta U = \delta Q - \delta W \tag{8.68}$$

The system under consideration in this section is a representative volume element (RVE), which is the smallest volume element that contains sufficient features of the microstructure and irreversible processes, such as damage and plasticity, to be representative of the material as a whole. Further discussions about the RVE can be found in Chapter 6.

In rate form, (8.68) is

$$\dot{U} = \dot{Q} - \dot{W} \tag{8.69}$$

where

$$\dot{U} = \frac{d}{dt} \int\limits_{\Omega} \rho u dV \tag{8.70}$$

Here ρ is the density, Ω is the volume of the RVE, and u is the *internal energy density*, which is an internal variable and a potential function.[7]

For a deformable solid, the rate of work done *by* the system is minus the product of the stress *applied* on the system times the rate of strain

$$\dot{W} = - \int\limits_{\Omega} \boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}} \, dV \tag{8.71}$$

where $\boldsymbol{\varepsilon}$ is the *total* strain (see (8.125)).

The heat flow into the RVE is given by

$$\dot{Q} = \int\limits_{\Omega} \rho r dV - \int\limits_{\partial\Omega} \mathbf{q} \cdot \mathbf{n} d\mathbf{A} \tag{8.72}$$

where r is the radiation heat per unit mass, $\mathbf{q}$ is the heat flow vector per unit area, and $\mathbf{n}$ is the outward normal vector to the surface $\partial\Omega$ enclosing the volume Ω. Since the volume Ω of the RVE does not change with time, and using the divergence theorem,[8] the *first law at the local level* becomes

$$\rho \dot{u} = \boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}} + \rho r - \nabla \cdot \mathbf{q} \tag{8.73}$$

The internal energy accounts for all the energy stored into the system. For example, a system undergoing elastic deformation $\delta\boldsymbol{\varepsilon}^e$, raising temperature δT, and damage in the form of cracks of area growing by δA_c undergoes a change of internal energy u given[9] by

$$\delta u = \boldsymbol{\sigma} : \delta\boldsymbol{\varepsilon}^e + C_p \delta T - (G - G_c)\delta A_c \tag{8.74}$$

[7] The values of the potential functions depend on the state and not on the path or process followed by the system to reach such state [88].

[8] $(\int_{\partial\Omega} \mathbf{q} \cdot \mathbf{n} dA = \int_{\Omega} \nabla \cdot \mathbf{q} dV)$; $div(\mathbf{q}) = \nabla \cdot \mathbf{q} = \partial \mathbf{q}_i / \partial x_i$

[9] Thermodynamics custom and [89] are followed here in representing the internal energy with the letter u, not to be confused with the displacement vector $\mathbf{u}$ used elsewhere.

where G is the strain energy release rate, G_c is the surface energy needed to create the increment of the two surfaces of an advancing crack, and $C_p = C_v$ is the specific heat capacity of the solid.

In general $\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}(\boldsymbol{\sigma}, u, s_\alpha)$ where s_α are internal variables. Let's assume for the time being that the system is adiabatic, i.e., $\rho r - \nabla \cdot \mathbf{q} = 0$. Further, if there are no dissipative effects or heat transfer, then u is a function of $\boldsymbol{\varepsilon}$ only, $u = u(\boldsymbol{\varepsilon}^e)$, where $\boldsymbol{\varepsilon}^e$ is the elastic strain. In such case, the internal energy density reduces to the strain energy density, which in rate form is

$$\dot{\varphi}(\boldsymbol{\varepsilon}) = \boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}}^e \tag{8.75}$$

and the complementary strain energy density is

$$\dot{\varphi}^*(\boldsymbol{\varepsilon}) = \boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}}^e - \dot{\varphi} = \dot{\boldsymbol{\sigma}} : \boldsymbol{\varepsilon}^e \tag{8.76}$$

8.3.2 Second Law

The *second law* of thermodynamics formalizes the fact that heat flows from hot to cold. Mathematically, the heat flow $\mathbf{q}$ has opposite direction to the gradient[10] of temperature T, which is formally written as

$$\mathbf{q} \cdot \nabla T \leq 0 \tag{8.77}$$

where the equal sign holds true only for adiabatic processes, i.e., when there is no heat exchange and thus no thermal irreversibility.

Let's visualize a process of heat transfer from a hot reservoir to a cold reservoir, happening in such a way that no heat is lost to, and no work is exchanged with, the environment. Once heat has flowed to the cold reservoir, it is impossible to transfer it back to the hot reservoir without adding external work. That is, the process of heat transfer is irreversible even though, on account of the first law energy balance (8.73), no energy has been lost. For future use (8.77) can be written[11] as

$$\mathbf{q} \cdot \nabla T^{-1} \geq 0 \tag{8.78}$$

The second law justifies the introduction of a new internal variable, the entropy density $s = s(u, \varepsilon)$, which is also a potential function [90]. According to the second law, the entropy density rate is $\dot{s} \geq 0$, where the equal sign holds true only for adiabatic processes.

Assume the specific entropy $s = s(u, \varepsilon)$ is a potential function such that for a reversible process [90]

$$ds = \left(\frac{\delta q}{T}\right)_{rev} \tag{8.79}$$

with $\delta Q = \int_\Omega \rho \; \delta q \; d\Omega$, where $\delta q = r - \rho^{-1}\nabla \cdot \mathbf{q}$ is the heat input per unit mass, and $S = \int_\Omega s \; \rho \; d\Omega$ is the entropy. We use δ, not d, to emphasize that δq is not the differential (perfect or total) of any (potential) function.

[10] The gradient of a scalar yields a vector, $\nabla T = \partial T / \partial x_i$

[11] $\nabla T^{-1} = -T^{-2} \nabla T$

As a preamble to the definition of conjugate variables (see (8.86, 8.92, 8.99)), note that using (8.79), the first law can be rewritten for a *reversible* process on an ideal gas ($pv = RT$), as the *Gibbs* equation for an ideal gas,

$$du = T\ ds - p\ dv \tag{8.80}$$

where v is the specific volume (volume per unit mass). It can be seen in (8.80) that v is conjugate to $-p$ for calculating work input for an ideal gas and s is conjugate to T for calculating thermal energy input.

For a *cyclic reversible* process returning to its initial state characterized by state variables (e.g., u, T, ε), by virtue of (8.79) we have $\oint ds = \oint \left(\frac{\delta \mathbf{q}}{T}\right)_{rev} = 0$. Since s is a potential function but q is not, for an *irreversible* process we have $\oint ds = 0$ but $\oint \left(\frac{\delta q}{T}\right)_{irrev} < 0$, as corroborated by experiments. The heat δq entering at temperature T_i provides less entropy input $\delta q/T_i$ than the entropy output $\delta q/T_o$ leaving the same cycle at temperature $T_o < T_i$ (see also [91, Example 6-2]). Since entropy is a potential function, and therefore a state variable, it always satisfies $\oint ds = 0$. Therefore, a negative net entropy supply must be compensated by internal entropy production. The entropy of a system can be raised or lowered by adding or extracting heat (in the form of $\delta q/T$) *but it is always raised by internal irreversible processes such as crack formation and so on* (positive dissipation principle).

Adiabatic systems do not exchange heat with the surroundings ($\delta q = 0$), so the only change in entropy is due to internal irreversibility $\dot{s} \geq 0$, where the equal sign holds for reversible processes only. Note that any system and its surroundings can be made adiabatic by choosing sufficiently large surroundings, e.g., the universe. For an arbitrary system, the total entropy rate is greater than (or equal to) the net entropy input due to heat

$$\dot{s} \geq \frac{r}{T} - \frac{1}{\rho}\nabla \cdot \left(\frac{\mathbf{q}}{T}\right) \tag{8.81}$$

The left-hand side of (8.81) represents the total entropy rate of the system. The right-hand side of (8.81) represents the external entropy supply rate. The difference is the internal entropy production rate

$$\dot{\gamma}_s = \dot{s} - \frac{r}{T} + \frac{1}{\rho}\nabla \cdot \left(\frac{\mathbf{q}}{T}\right) \geq 0 \tag{8.82}$$

Equation 8.82 is called the *local Clausius-Duhem inequality.* Noting that $\nabla \cdot \left(T^{-1}\mathbf{q}\right) = T^{-1}\nabla \cdot \mathbf{q} + \mathbf{q}\nabla\, T^{-1}$ results in

$$\dot{\gamma}_s = \dot{s} - \frac{1}{\rho T}\left(\rho r - \nabla \cdot \mathbf{q}\right) + \frac{1}{\rho}\,\mathbf{q} \cdot \nabla\, T^{-1} \geq 0 \tag{8.83}$$

where the first two terms represent the local entropy production due to local dissipative phenomena, and the last term represents the entropy production due to heat conduction[12] [90]. Assuming it is possible to identify all local dissipative phenom-

[12] Even absent local dissipative phenomena, $\mathbf{q} \cdot \nabla T^{-1} \geq 0$ represents the well-known fact that heat flows opposite to the temperature gradient ∇T.

ena, their contributions can be written as products of conjugate variables $p_\alpha \dot{s_\alpha} \geq 0$, and (8.83) can be written as

$$\rho T \dot{\gamma}_s = p_\alpha \dot{s_\alpha} + T\mathbf{q} \cdot \nabla T^{-1} \geq 0 \tag{8.84}$$

where $\alpha = 1 \ldots n$, spans the total number of dissipative phenomena considered. Note that the dissipation is a scalar given by the contracted product of a thermodynamic force p_α times the increment of a measurable state variable s_α. The state variable, also called thermodynamic flux, describes univocally the effects of history (e.g., yield, damage) on the material. Note that γ_s is defined as an entropy, not as a dissipation heat, so that it is a potential function, while q is not.

For the particular case of damage due to penny-shaped cracks growing self-similarly [63], the state variable is the crack area A_c and the thermodynamic force is the energy available to grow the cracks $p_c = G - G_c$, which is equal to the difference between the energy release rate (ERR) G and the critical ERR $G_c = 2\gamma_c$, the latter being equal to twice the surface energy because two surfaces must be created every time a crack appears (see Chapter 10). In this case, the dissipation (heat) is $\rho T \dot{\gamma} = p_c \dot{A}_c$.

From the first law (8.73), considering an adiabatic process ($\rho r - \nabla \cdot \mathbf{q}$) and using the chain rule $\dot{u} = \partial u / \partial \boldsymbol{\varepsilon} : \dot{\boldsymbol{\varepsilon}}$ we have

$$\left[\boldsymbol{\sigma} - \rho \frac{\partial u}{\partial \boldsymbol{\varepsilon}} \right] : \dot{\boldsymbol{\varepsilon}} = 0 \tag{8.85}$$

Since $\dot{\boldsymbol{\varepsilon}} = 0$ would be a trivial solution, the stress tensor, conjugate to strain, is defined as

$$\boldsymbol{\sigma} = \rho \frac{\partial u}{\partial \boldsymbol{\varepsilon}} \tag{8.86}$$

The Clausius-Duhem inequality (8.83) for an *isothermal* ($\nabla T = 0$) system reduces to

$$\dot{\gamma}_s = \dot{s} - \frac{1}{\rho T} \left(\rho r - \nabla \cdot \mathbf{q} \right) \geq 0 \tag{8.87}$$

and using the first law we get

$$\rho T \dot{\gamma}_s = \rho T \dot{s} - \left(\rho \dot{u} - \boldsymbol{\sigma} : \boldsymbol{\varepsilon} \right) \geq 0 \tag{8.88}$$

The *Helmholtz free energy (HFE) density* is defined as

$$\psi(T, \boldsymbol{\varepsilon}, s_\alpha) = u - Ts \tag{8.89}$$

which is also a potential function. The corresponding extensive function is the Helmholtz free energy[13] $A = \int_\Omega \rho \psi dV$. The rate of change of HFE density is

$$\dot{\psi} = \dot{u} - \dot{T}s - T\dot{s} \tag{8.90}$$

[13] The nomenclature of [89] has been used.

and introducing (8.88), with $\dot{\gamma}_s = 0$ at an equilibrium state, we get

$$\rho\dot{\psi} = -\rho s\dot{T} + \boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}} \tag{8.91}$$

from which an alternative definition of stress, conjugate to strain, is found as

$$\boldsymbol{\sigma} = \rho\frac{\partial\psi}{\partial\boldsymbol{\varepsilon}} = \mathbf{C} : \varepsilon \tag{8.92}$$

where the secant elastic stiffness, which is affected by dissipative phenomena, including damage, is defined as

$$\mathbf{C}(s_\alpha) = \rho\frac{\partial^2\psi}{\partial\boldsymbol{\varepsilon}^2} \tag{8.93}$$

Using the first law (8.73) in the internal entropy production per unit volume, or local Clausius-Duhem inequality (8.83), and expanding $\nabla\cdot(\mathbf{q}T^{-1}) = T^{-1}\nabla\cdot\mathbf{q} + \mathbf{q}\cdot\nabla T^{-1}$, we get

$$\rho T\dot{\gamma}_s = \frac{\mathbf{q}}{T}\cdot\nabla T^{-1} - \rho\left(\dot{\psi} + s\dot{T} - \rho^{-1}\boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}}\right) \geq 0 \tag{8.94}$$

Realizing that $\nabla T^{-1} = -\nabla T/T^2$, the Clausius-Duhem inequality becomes

$$T\rho\dot{\gamma}_s = \boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}} - \rho\left(\dot{\psi} + s\dot{T}\right) - \frac{\mathbf{q}}{T}\cdot\nabla T \geq 0 \tag{8.95}$$

Since the HFE density is a function of the internal variables $\boldsymbol{\varepsilon}, T, s_\alpha$, we have

$$\dot{\psi} = \left.\frac{\partial\psi}{\partial\boldsymbol{\varepsilon}}\right|_{T,s_\alpha} : \dot{\boldsymbol{\varepsilon}} \;+\; \left.\frac{\partial\psi}{\partial T}\right|_{\boldsymbol{\varepsilon},s_\alpha} \dot{T} \;+\; \left.\frac{\partial\psi}{\partial s_\alpha}\right|_{\boldsymbol{\varepsilon},T} \dot{s}_\alpha \quad ; \quad \alpha = 1\ldots n \tag{8.96}$$

where $\left.\dfrac{\partial}{\partial y}\right|_x$ represents the partial derivative with respect to y at constant x.

Inserting (8.96) into (8.95), using (8.89), (8.92), and $\nabla T^{-1} = -\nabla T/T^2$, the second law can be written as follows

$$\dot{\gamma} = \rho T\dot{\gamma}_s = -\rho\frac{\partial\psi}{\partial s_\alpha}s_\alpha + T\mathbf{q}\cdot\nabla T^{-1} \geq 0 \tag{8.97}$$

where $\dot{\gamma}$ is the heat dissipation rate per unit volume. Comparing (8.97) to (8.84) it becomes clear that $-\rho\partial\psi/\partial s_\alpha = p_\alpha$ are the thermodynamic forces conjugated to s_α, which provides a definition for the thermodynamic forces.

The complementary free-energy density, or *Gibbs energy density*, is defined as

$$\chi = \rho^{-1}\boldsymbol{\sigma} : \boldsymbol{\varepsilon} - \psi \tag{8.98}$$

which is also a potential function. The corresponding extensive function is the Gibbs energy[14] $G = \int_\Omega \rho\chi dV$. From (8.98) it follows the definition of strain, conjugate

[14] The nomenclature of [89] has been used.

to stress, and the definition of the thermodynamic forces, conjugate to the state variables s_α, as

$$\varepsilon = \rho\frac{\partial\chi}{\partial\boldsymbol{\sigma}} \quad ; \quad p_\alpha = \rho\frac{\partial\chi}{\partial s_\alpha} = -\rho\frac{\partial\psi}{\partial s_\alpha} \tag{8.99}$$

where s_α includes the damage variables and consequently p_α includes the thermodynamic damage forces (see Example 8.4, p. 311).

The secant elastic compliance, which is affected by dissipative phenomena, including damage, is defined by

$$S(s_\alpha) = \rho\frac{\partial^2\chi}{\partial\boldsymbol{\sigma}^2} \tag{8.100}$$

Example 8.4 *The following Gibbs free energy is proposed to represent the onset and accumulation of transverse matrix cracks resulting from transverse tension and in-plane shear loads:*

$$\chi = \frac{1}{2\,\rho}\left[\frac{\sigma_1^2}{\widetilde{E}_1} + \frac{\sigma_2^2}{\left(1-D_2\right)^2\widetilde{E}_2} + \frac{\sigma_6^2}{\left(1-D_6\right)^2 2G_{12}} - \left(\frac{\widetilde{\nu}_{21}}{\widetilde{E}_2} + \frac{\widetilde{\nu}_{12}}{\widetilde{E}_1}\right)\frac{\sigma_1\sigma_2}{1-D_2}\right]$$

where $\widetilde{E}_1$, $\widetilde{E}_2$, $\widetilde{\nu}_{12}$, $\widetilde{\nu}_{21}$, and $\widetilde{G}_{12}$ are the undamaged in-plane elastic orthotropic properties of a unidirectional lamina where the subscript $()_1$ denotes the fiber direction and $()_2$ denotes the transverse direction. The damage variables D_2 and D_6 represent the effect of matrix cracks. The proposed distinguishes between active (D_{2+}) and passive damage (D_{2-}), corresponding to the opening or closure of transverse matrix cracks, respectively. The determination of the active damage variable is based on the following equation:

$$D_2 = D_{2+}\frac{\langle\sigma_2\rangle}{|\sigma_2|} + D_{2-}\frac{\langle-\sigma_2\rangle}{|\sigma_2|}$$

where $\langle x\rangle$ is defined as $\langle x\rangle := \frac{1}{2}\left(x+|x|\right)$.

For a lamina in a state of plane stress, subjected to in-plane stress only, without fiber damage ($D_1 = 0$), and using the energy equivalence principle (8.8), derive expressions for a) the secant stiffness tensor, b) the effective stress, and c) the thermodynamic forces associated to the model. Use tensor components of strain ($\varepsilon_1, \varepsilon_2, \varepsilon_6$).

Solution to Example 8.4 *The constitutive model is defined as the derivative of the Gibbs free energy with respect to the stress tensor*

$$\boldsymbol{\varepsilon} = \rho\frac{\partial\chi}{\partial\boldsymbol{\sigma}} = \mathbf{S} : \boldsymbol{\sigma}$$

whith $\varepsilon_6 = \gamma_6/2$ and the compliance tensor $\mathbf{S}$ defined as

$$\mathbf{S} = \rho\frac{\partial^2\chi}{\partial\boldsymbol{\sigma}^2}$$

The compliance tensor for plane stress $\mathbf{S}$ in Voigt contracted notation is

$$\mathbf{S} = \begin{bmatrix} \dfrac{1}{\widetilde{E}_1} & -\dfrac{\widetilde{\nu}_{21}}{\widetilde{E}_2\left(1-D_2\right)} & 0 \\ -\dfrac{\widetilde{\nu}_{12}}{\widetilde{E}_1\left(1-D_2\right)} & \dfrac{1}{\widetilde{E}_2\left(1-D_2\right)^2} & 0 \\ 0 & 0 & \dfrac{1}{2\widetilde{G}_{12}\left(1-D_6\right)^2} \end{bmatrix}$$

The damage variables appear in S_{12}, S_{21}, S_{22}, and S_{66}. To perform tensor products using matrix multiplications, see (A.14) and (A.20). Using the energy equivalence principle and (8.67), the compliance matrix can be written as

$$\mathbf{S} = \mathbf{M}^{-1} : \widetilde{\mathbf{S}} : \mathbf{M}^{-1}$$

where the undamaged compliance is

$$\widetilde{\mathbf{S}} = \begin{bmatrix} \dfrac{1}{\widetilde{E}_1} & -\dfrac{\widetilde{\nu}_{21}}{\widetilde{E}_2} & 0 \\ -\dfrac{\widetilde{\nu}_{12}}{\widetilde{E}_1} & \dfrac{1}{\widetilde{E}_2} & 0 \\ 0 & 0 & \dfrac{1}{2\widetilde{G}_{12}} \end{bmatrix}$$

and where the effective damage tensor $\mathbf{M}$, *written in contracted notation, multiplied by the 3×3 version of the Reuter matrix (A.13) is*

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & (1 - D_2) & 0 \\ 0 & 0 & (1 - D_6) \end{bmatrix}$$

The stiffness tensor $\mathbf{C}$ *is obtained by*

$$\mathbf{C} = \mathbf{M} : \widetilde{\mathbf{C}} : \mathbf{M}$$

where, for this particular example, the secant stiffness tensor is

$$\mathbf{C} = \begin{bmatrix} \dfrac{\widetilde{E}_1}{1 - \widetilde{\nu}_{21}\widetilde{\nu}_{12}} & \dfrac{\widetilde{\nu}_{12}\widetilde{E}_2 \, (1 - D_2)}{1 - \widetilde{\nu}_{21}\widetilde{\nu}_{12}} & 0 \\ \dfrac{\widetilde{\nu}_{21}\widetilde{E}_1 \, (1 - D_2)}{1 - \widetilde{\nu}_{21}\widetilde{\nu}_{12}} & \dfrac{\widetilde{E}_2 \, (1 - D_2)^2}{1 - \widetilde{\nu}_{21}\widetilde{\nu}_{12}} & 0 \\ 0 & 0 & 2\widetilde{G}_{12} \left(1 - \widetilde{D}_6\right)^2 \end{bmatrix}$$

The effective stress $\widetilde{\boldsymbol{\sigma}}$ *is related to the nominal stress* $\boldsymbol{\sigma}$ *by the effective damage tensor* $\mathbf{M}$ *using* $\widetilde{\boldsymbol{\sigma}} = \mathbf{M}^{-1} : \boldsymbol{\sigma}$, *which yields*

$$\widetilde{\boldsymbol{\sigma}}^T = \left\{ \sigma_1, \frac{\sigma_2}{1 - D_2}, \frac{\sigma_6}{1 - D_6} \right\}$$

The thermodynamic forces are obtained by using $Y = \rho \partial \chi / \partial D$, *which for this particular example yield*

$$\mathbf{Y} = \left\{ \begin{array}{c} Y_1 \\ Y_2 \\ Y_6 \end{array} \right\} = \left\{ \begin{array}{c} 0 \\ \dfrac{{\sigma_2}^2}{(1 - D_2)^3 \, \widetilde{E}_2} - \dfrac{\sigma_1 \sigma_2 \widetilde{\nu}_{12}}{(1 - D_2)^2 \, \widetilde{E}_1} \\ \dfrac{{\sigma_6}^2}{(1 - D_6)^3 \, 2\widetilde{G}_{12}} \end{array} \right\}$$

8.4 Kinetic Law in Three-Dimensional Space

The damage variable $\mathbf{D}$ introduced in Section 8.2 is a state variable that represents the history of what happened to the material. Next, a kinetic equation is needed to predict the evolution of damage in terms of the thermodynamic forces. Kinetic equations can be written directly in terms of internal variables as in (8.21) or as derivatives of potential functions. For three-dimensional problems, it is convenient to derive the kinetic law from a potential function, similar to the flow potential used in plasticity theory.

Two functions are needed. A damage surface $g(\mathbf{Y}(\mathbf{D}), \gamma(\delta)) = 0$ and a convex damage potential $f(\mathbf{Y}(\mathbf{D}), \gamma(\delta)) = 0$ are postulated. The damage surface delimits a region in the space of thermodynamic forces $\mathbf{Y}$ where damage does not occur because the thermodynamic force $\mathbf{Y}$ is inside the g-surface. The function $\gamma(\delta)$ accomplishes the expansion of g and f needed to model hardening. The damage potential controls the direction of damage evolution (8.102).

If the damage surface and the damage potential are identical ($g = f$), the model is said to be associated and the computational implementation is simplified significantly. For convenience, the damage surface is assumed to be separable in the variables $\mathbf{Y}$ and γ, and written as the sum (see (8.11)-(8.12))

$$g(\mathbf{Y}(\mathbf{D}), \gamma(\delta)) = \hat{g}(\mathbf{Y}(\mathbf{D})) - (\gamma(\delta) + \gamma_0) \tag{8.101}$$

where $\mathbf{Y}$ is the thermodynamic force tensor, $\gamma(\delta)$ is the hardening function, γ_0 is the damage threshold, and δ is the hardening variable.

As a result of damage, $\hat{g}$ may grow but the condition $g < 0$ must be satisfied. This is possible by *increasing* the value of the hardening function $\gamma(\delta)$, effectively allowing $\hat{g}(\mathbf{Y}(\mathbf{D}))$ to grow. Formally, the hardening function $\gamma(\delta)$ can be derived from the dissipation potential as per (8.99), (8.124), provided the form of the potential can be inferred from knowledge about the hardening process. Alternatively, the form of the function (e.g., polynomial, Prony series, etc.) can be chosen so that the complete model fits adequately the experimental data available. The latter approach is more often followed in the literature.

When $g = 0$, damage occurs, and a kinetic equation is needed to determine the magnitude and components of the damage $\dot{\mathbf{D}}$. This is accomplished by

$$\dot{\mathbf{D}} = \frac{\partial \mathbf{D}}{\partial \mathbf{Y}} = \dot{\lambda} \frac{\partial f}{\partial \mathbf{Y}} \tag{8.102}$$

where $\dot{\lambda}$ yields the magnitude of the damage increment and $\partial f / \partial \mathbf{Y}$ is a direction in $\mathbf{Y}$-space. To find the damage multiplier $\dot{\lambda}$, it is postulated that $\dot{\lambda}$ is also involved in the determination of the rate of change of the hardening variable as follows

$$\dot{\delta} = \dot{\lambda} \frac{\partial g}{\partial \gamma} \tag{8.103}$$

There are two possible situations regarding g and $\dot{\lambda}$:

i. If $g < 0$, damage is not growing and $\dot{\lambda} = 0$, so $\dot{\mathbf{D}} = 0$.

ii. If $g = 0$, damage occurs and $\dot{\lambda} > 0$, so $\dot{\mathbf{D}} > 0$.

These are summarized by the Kuhn-Tucker conditions

$$\dot{\lambda} \geq 0 \quad ; \quad g \leq 0 \quad ; \quad \dot{\lambda} g = 0 \tag{8.104}$$

The value of $\dot{\lambda}$ can be determined by the consistency condition, which leads to

$$\dot{g} = \frac{\partial g}{\partial \mathbf{Y}} : \dot{\mathbf{Y}} + \frac{\partial g}{\partial \gamma}\dot{\gamma} = 0 \; ; \quad g = 0 \tag{8.105}$$

On the other hand, the rates of thermodynamic forces and hardening function can be written as

$$\begin{aligned} \dot{\mathbf{Y}} &= \frac{\partial \mathbf{Y}}{\partial \boldsymbol{\varepsilon}} : \dot{\boldsymbol{\varepsilon}} + \frac{\partial \mathbf{Y}}{\partial \mathbf{D}} : \dot{\mathbf{D}} \\ \dot{\gamma} &= \frac{\partial \gamma}{\partial \delta}\dot{\delta} \end{aligned} \tag{8.106}$$

or in function of $\dot{\lambda}$, introducing (8.103) and (8.104) into (8.106) as follows

$$\begin{aligned} \dot{\mathbf{Y}} &= \frac{\partial \mathbf{Y}}{\partial \boldsymbol{\varepsilon}} : \dot{\boldsymbol{\varepsilon}} + \dot{\lambda}\frac{\partial \mathbf{Y}}{\partial \mathbf{D}} : \frac{\partial f}{\partial \mathbf{Y}} \\ \dot{\gamma} &= \frac{\partial \gamma}{\partial \delta}\dot{\lambda}\frac{\partial g}{\partial \gamma} \end{aligned} \tag{8.107}$$

Introducing (8.107) into (8.105) we obtain the following equation

$$\dot{g} = \frac{\partial g}{\partial \mathbf{Y}} : \left[\frac{\partial \mathbf{Y}}{\partial \boldsymbol{\varepsilon}} : \dot{\boldsymbol{\varepsilon}} + \dot{\lambda}\frac{\partial \mathbf{Y}}{\partial \mathbf{D}} : \frac{\partial f}{\partial \mathbf{Y}}\right] + \frac{\partial g}{\partial \gamma}\frac{\partial \gamma}{\partial \delta}\dot{\lambda}\frac{\partial g}{\partial \gamma} = 0 \tag{8.108}$$

Next, $\partial f/\partial \gamma = \partial g/\partial \gamma = -1$, (8.108) can be written as

$$\dot{g} = \frac{\partial g}{\partial \mathbf{Y}} : \frac{\partial \mathbf{Y}}{\partial \boldsymbol{\varepsilon}} : \dot{\boldsymbol{\varepsilon}} + \left[\frac{\partial g}{\partial \mathbf{Y}} : \frac{\partial \mathbf{Y}}{\partial \mathbf{D}} : \frac{\partial f}{\partial \mathbf{Y}} + \frac{\partial \gamma}{\partial \delta}\right]\dot{\lambda} = 0 \tag{8.109}$$

Therefore, the damage multiplier $\dot{\lambda}$ can be obtained as

$$\dot{\lambda} = \begin{cases} \mathbf{L}^d : \dot{\boldsymbol{\varepsilon}} & \text{when} \quad g = 0 \\ 0 & \text{when} \quad g < 0 \end{cases} \tag{8.110}$$

where

$$\mathbf{L}^d = -\frac{\dfrac{\partial g}{\partial \mathbf{Y}} : \dfrac{\partial \mathbf{Y}}{\partial \boldsymbol{\varepsilon}}}{\dfrac{\partial g}{\partial \mathbf{Y}} : \dfrac{\partial \mathbf{Y}}{\partial \mathbf{D}} : \dfrac{\partial f}{\partial \mathbf{Y}} + \dfrac{\partial \gamma}{\partial \delta}} \tag{8.111}$$

Equations (8.103), (8.104), and (8.110) yield the pair $\mathbf{D}, \delta$ in rate form as

$$\dot{\mathbf{D}} = \mathbf{L}^d : \frac{\partial f}{\partial Y} : \dot{\boldsymbol{\varepsilon}} \quad ; \quad \dot{\delta} = -\dot{\lambda} \tag{8.112}$$

The tangent constitutive equation can be obtained by differentiation of the constitutive equation $\boldsymbol{\sigma} = \mathbf{C} : \boldsymbol{\varepsilon}$, which yields

$$\dot{\sigma} = \mathbf{C} : \dot{\varepsilon} + \dot{\mathbf{C}} : \boldsymbol{\varepsilon} \tag{8.113}$$

where the last term represents the stiffness reduction. Next, the last term in (8.113) can be written as

$$\dot{C} : \boldsymbol{\varepsilon} = \frac{\partial \mathbf{C}}{\partial \mathbf{D}} : \dot{\mathbf{D}} : \boldsymbol{\varepsilon} \tag{8.114}$$

Introduce (8.112) and rearrange

$$\dot{C} : \boldsymbol{\varepsilon} = \frac{\partial \mathbf{C}}{\partial \mathbf{D}} : \boldsymbol{\varepsilon} : \mathbf{L}^d : \frac{\partial f}{\partial \mathbf{Y}} : \dot{\boldsymbol{\varepsilon}} \tag{8.115}$$

Since $(\boldsymbol{\varepsilon}, \mathbf{D})$ are state variables, and thus independent variables,

$$\frac{\partial \boldsymbol{\varepsilon}}{\partial \mathbf{D}} = 0 \tag{8.116}$$

Therefore,

$$\dot{\mathbf{C}} : \boldsymbol{\varepsilon} = \frac{\partial \boldsymbol{\sigma}}{\partial \mathbf{D}} : \mathbf{L}^d : \frac{\partial f}{\partial \mathbf{Y}} : \dot{\boldsymbol{\varepsilon}} \tag{8.117}$$

Finally, reintroduce the above into (8.113) to get

$$\dot{\boldsymbol{\sigma}} = \mathbf{C}^{ed} : \dot{\boldsymbol{\varepsilon}} \tag{8.118}$$

where the *damaged tangent constitutive tensor*, $\mathbf{C}^{ed}$, can be written as follows

$$\mathbf{C}^{ed} = \begin{cases} \mathbf{C} & \text{if } \dot{\mathbf{D}} \leq 0 \\ \mathbf{C} + \dfrac{\partial \boldsymbol{\sigma}}{\partial \mathbf{D}} : \mathbf{L}^d : \dfrac{\partial f}{\partial \mathbf{Y}} & \text{if } \dot{\mathbf{D}} \geq 0 \end{cases} \tag{8.119}$$

The internal variables $\mathbf{D}, \delta$, and related variables, are found using numerical integration, usually using a return-mapping algorithm as explained in Section 8.4.1.

As explained in Section 8.1.3 and 8.4, a number of internal material parameters are needed to define the damage surface, damage potential, and hardening functions. These parameters cannot be obtained directly from simple tests, but rather the model is *identified* by adjusting the internal parameters in such a way that model predictions fit well some observed behavior that can be quantified experimentally. Model identification is very specific to the particular model formulation, material, availability of experiments, and feasibility of conducting relevant experiments. Therefore, model identification can be explained only on a case by case basis, as it is done in Example 8.3, p. 296.

8.4.1 Return-Mapping Algorithm

A return-mapping algorithm [82,84,85] is used to solve for the variables $\dot{\lambda}$, $\dot{\delta}$, $\dot{\mathbf{D}}$, δ, and $\mathbf{D}$, in numerically approximated form.

The internal variables are updated by a linearized procedure between two consecutive iterations ($k-1$ and k). The first-order linearization of (8.109) yields

$$(g)_k - (g)_{k-1} = \left(\frac{\partial g}{\partial \mathbf{Y}} : \frac{\partial \mathbf{Y}}{\partial \mathbf{D}} : \frac{\partial f}{\partial \mathbf{Y}} + \frac{\partial \gamma}{\partial \delta}\right)_{k-1} \Delta\lambda_k = 0 \tag{8.120}$$

Successful iterations yield $[g]_k = 0$ and

$$\Delta\lambda_k = \frac{-(g)_{k-1}}{\left(\frac{\partial g}{\partial \mathbf{Y}} : \frac{\partial \mathbf{Y}}{\partial \mathbf{D}} : \frac{\partial f}{\partial \mathbf{Y}} + \frac{\partial \gamma}{\partial \delta}\right)_{k-1}} \tag{8.121}$$

The complete algorithm used for a typical integration of constitutive equations is shown next:

i. Retrieve the strain $(\boldsymbol{\varepsilon})^{n-1}$ from the previous increment and the strain increment $(\Delta\boldsymbol{\varepsilon})^n$ for the current increment from the FEM code. The updated strain is calculated as

$$(\boldsymbol{\varepsilon})^n = (\boldsymbol{\varepsilon})^{n-1} + (\Delta\varepsilon)^n$$

ii. Retrieve the state variables from the previous step and start the return-mapping algorithm by setting the predictor iteration $k = 0$

$$(\mathbf{D})_0^n = (\mathbf{D})^{n-1}; \quad (\delta)_0^n = (\delta)^{n-1}$$

iii. Update the secant stiffness and Cauchy stress, which are used to calculate the thermodynamic forces and damage hardening at this point

$$\begin{array}{l} (\mathbf{C})_k^n = (\mathbf{M})_k^n : \widetilde{\mathbf{C}} : (\mathbf{M})_k^n \\ (\sigma)_k^n = (\mathbf{C})_k^n : (\epsilon)^n \\ (\mathbf{Y})_k^n \quad ; \quad (\gamma)_k^n \end{array}$$

iv. The damage threshold is evaluated at this point

$$(g)_k = g\left((\mathbf{Y})_k^n, (\gamma(\delta))_k^n, \gamma_0\right)$$

There are two possible cases:

(a) If $(g)_k \leq 0$, there is no damage, then $\Delta\lambda_k = 0$. Go to (viii).

(b) If $(g)_k > 0$, there is damage evolution, then $\Delta\lambda_k > 0$. Go to (v).

v. Damage evolution. Starting at iteration k, the damage multiplier is found from $(g)_k = 0$ as

$$\Delta\lambda_k = \frac{-(g)_{k-1}}{\left(\frac{\partial g}{\partial \mathbf{Y}}\right)_{k-1} : \left(\frac{\partial \mathbf{Y}}{\partial \mathbf{D}}\right)_{k-1} : \left(\frac{\partial f}{\partial \mathbf{Y}}\right)_{k-1} + \left(\frac{\partial \gamma}{\partial \delta}\right)_{k-1}}$$

vi. Update the state variables using $\Delta\lambda_k$

$$\left(D_{ij}\right)_k^n = \left(D_{ij}\right)_{k-1}^n + \Delta\lambda_k \left(\frac{\partial f}{\partial \mathbf{Y}}\right)_{k-1}$$

$$(\delta)_k^n = (\delta)_{k-1}^n + \Delta\lambda_k \left(\frac{\partial f}{\partial \gamma}\right)_{k-1} = (\delta)_{k-1}^n - \Delta\lambda_k$$

vii. End of linearized damage process. Go to (iii).

viii. Compute the tangent stiffness tensor

$$\left(\mathbf{C}^{ed}\right)^n \quad = (\mathbf{C})^n + \left(\frac{\partial \sigma}{\partial \mathbf{D}}\right)^n : \left(\mathbf{L}^d\right)^n : \left(\frac{\partial f}{\partial \mathbf{Y}}\right)^n$$

ix. Store the stress and state variables to be used on the next load increment

$$(\boldsymbol{\sigma})^n = (\boldsymbol{\sigma})_k^n\,; \quad (\mathbf{D})^n = (\mathbf{D})_k^n\,; \quad (\delta)^n = (\delta)_k^n$$

x. End of the integration algorithm.

Example 8.5 *Implement the damage model developed in Example 8.4 (p. 311) into a user material subroutine. Use a return mapping algorithm as shown in Section 8.4.1. Furthermore, use the following damage activation function*

$$g = \hat{g} - \hat{\gamma} = \sqrt{\left(1 - \frac{G_{Ic}}{G_{IIc}}\right)\frac{Y_2\widetilde{E}_2}{{F_{2t}}^2} + \frac{G_{Ic}}{G_{IIc}}\left(\frac{Y_2\widetilde{E}_2}{{F_{2t}}^2}\right)^2 + \left(\frac{Y_6\widetilde{G}_{12}}{{F_6}^2}\right)^2} - \hat{\gamma} \leq 0$$

where G_{Ic} and G_{IIc} are the critical energy release in mode I and in mode II, respectively, and F_{2t} and F_6 are the transverse tensile strength and the shear strength, respectively. Also, use the following damage hardening function

$$\hat{\gamma} = \gamma + \gamma_0 = c_1\left[\exp\left(\frac{\delta}{c_2}\right) - 1\right] + \gamma_0 \quad ; \quad \gamma_0 - c_1 \leq \widehat{\gamma} \leq \gamma_0$$

where γ_0 defines the initial threshold value, and c_1 and c_2 are material parameters. For this particular damage model, the model parameters for AS4/8852 carbon/epoxy are given in Tables 8.1 and 8.2.

Table 8.1: Elastic and strength properties for AS4/8852 unidirectional lamina.

$\widetilde{\mathbf{E}}_\mathbf{1}$	$\widetilde{\mathbf{E}}_\mathbf{2}$	$\widetilde{\mathbf{G}}_\mathbf{12}$	$\widetilde{\nu}_\mathbf{12}$	$\mathbf{F_{2t}}$	$\mathbf{F_6}$
171.4 GPa	9.08 GPa	5.29 GPa	0.32	62.29 MPa	92.34 MPa

Table 8.2: Critical energy release, and hardening parameters for AS4/8852 unidirectional lamina.

$\mathbf{G_{Ic}}$	$\mathbf{G_{IIc}}$	$\gamma_\mathbf{0}$	$\mathbf{c_1}$	$\mathbf{c_2}$
170 J/m^2	230 J/m^2	1.0	0.5	-1.8

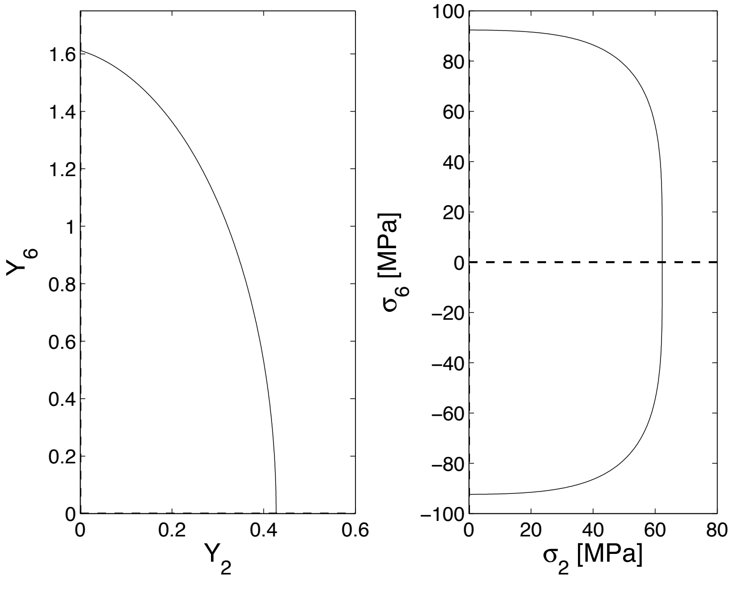


Fig. 8.8: Initial damage surface in thermodynamic and stress spaces.

Solution to Example 8.5 *This model represents damage caused by both transverse tensile stress and in-plane shear stress. Longitudinal tension/compression have no effect. Therefore, the model is defined in the thermodynamic force space Y_2, Y_6. The shape of the damage surface for AS4/8852 lamina is shown in Figure 8.8.*

To implement the return mapping algorithm shown in Section 8.4.1, expressions for $\partial f/\partial \mathbf{Y}$, $\partial g/\partial \mathbf{Y}$, $\partial f/\partial \gamma$, $\partial g/\partial \gamma$,$\partial \gamma/\partial \delta$, and $\partial \mathbf{Y}/\partial \mathbf{D}$ are needed.

Assuming $f = g$, the derivative of the potential function and the damage surface with respect to (w.r.t) the thermodynamic forces is given by

$$\frac{\partial g}{\partial \mathbf{Y}} = \frac{\partial f}{\partial \mathbf{Y}} = \left\{ \begin{array}{c} 0 \\ \frac{1}{\hat{g}} \left(\left(1 - \frac{G_{Ic}}{G_{IIc}} \right) \frac{1}{4F_{2t}} \sqrt{\frac{2E_2}{Y_2}} + \frac{G_{Ic}}{G_{IIc}} \frac{E_2}{(F_{2t})^2} \right) \\ \frac{1}{\hat{g}} G_{12} \end{array} \right\}$$

and the derivative of the damage surface w.r.t. the damage hardening function is

$$\frac{\partial g}{\partial \gamma} = \frac{\partial f}{\partial \gamma} = -1$$

Also, the derivative of the hardening function γ w.r.t conjugate variable δ is needed

$$\frac{\partial \gamma}{\partial \delta} = \frac{c_1}{c_2} \exp\left(\frac{\delta}{c_2} \right)$$

Next, the derivative of the thermodynamic forces w.r.t the internal damage variables is written as

$$\frac{\partial \mathbf{Y}}{\partial \mathbf{D}} = \left. \frac{\partial \mathbf{Y}}{\partial \mathbf{D}} \right|_{\sigma = const} + \frac{\partial \mathbf{Y}}{\partial \boldsymbol{\sigma}} : \frac{\partial \boldsymbol{\sigma}}{\partial \mathbf{D}}$$

Furthermore, the derivative of the thermodynamic forces w.r.t strain is written as

$$\frac{\partial \mathbf{Y}}{\partial \boldsymbol{\varepsilon}} = \frac{\partial \mathbf{Y}}{\partial \boldsymbol{\sigma}} : \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}} = \frac{\partial \mathbf{Y}}{\partial \boldsymbol{\sigma}} : \mathbf{C}$$

The following are written in contracted notation as

$$\frac{\partial \mathbf{Y}}{\partial \boldsymbol{\sigma}} = \left[\begin{array}{ccc} 0 & 0 & 0 \\ \dfrac{-\sigma_2 \widetilde{\nu}_{12}}{(1-D_2)^2 \widetilde{E}_1} & \dfrac{2\sigma_2}{(1-D_2)^3 \widetilde{E}_2} - \dfrac{\sigma_1 \widetilde{\nu}_{12}}{(1-D_2)^2 \widetilde{E}_1} & 0 \\ 0 & 0 & \dfrac{\sigma_6}{(1-D_6)^3 \widetilde{G}_{12}} \end{array} \right]$$

and

$$\frac{\partial \boldsymbol{\sigma}}{\partial \mathbf{D}} = \left[\begin{array}{ccc} 0 & -\dfrac{\widetilde{E}_2 \widetilde{\nu}_{12}}{1 - \widetilde{\nu}_{12}\widetilde{\nu}_{21}} \varepsilon_2 & 0 \\ 0 & -\dfrac{\widetilde{E}_1 \widetilde{\nu}_{21}}{1 - \widetilde{\nu}_{12}\widetilde{\nu}_{21}} \varepsilon_1 - \dfrac{2\,(1-D_2)\widetilde{E}_2}{1 - \widetilde{\nu}_{12}\widetilde{\nu}_{21}} \varepsilon_2 & 0 \\ 0 & 0 & -4(1-D_6)\,\widetilde{G}_{12} \varepsilon_6 \end{array} \right]$$

The equations are implemented in the subroutine `umatps85.for`, which is available in [5]. The material model subroutine can be used in conjunction with plain stress elements and laminated shells elements. A pseudo code for transverse tensile load is shown below.

i. Setting the `Work Directory`

```
Menu:   File, Set Work Directory, [C:\SIMULIA\User\Ex_8.5], OK
Menu:   File, Save As, [C:\SIMULIA\User\Ex_8.5\Ex_8.5.tra.cae], OK
```

ii. Creating the part

```
Module: Part
Menu:   Part, Create
    2D Planar, Deformable, Shell, Cont
Menu:   Add, Line, Rectangle, [0,0], [10,10], X, Done
```

iii. Defining materials, sections, and assigning sections to parts

```
Module: Property
Menu:   Material, Create
    General, User Material
    # read material properties from a file
    # right-click in Data area, Read from file, [props.txt], OK, OK
    General, Depvar
    Number of solution-dependent state variables [3], OK
Menu:   Section, Create
    Solid, Homogeneous, Cont
    # checkmark: Plane stress/strain thickness [1.0], OK
Menu:   Assign, Section, # pick: part, Done, OK
```

iv. Creating the assembly

```
Module: Assembly
Menu:   Instance, Create, Independent, OK
```

v. Defining analysis steps

```
Module: Step
Menu:   Step, Create
    Name [transverse], Procedure type: General, Static/General, Cont
    Tab: Basic, Time Period [50]
    Tab: Incrementation, Type: Fixed
    Maximum number of increments [50], Increment size [2.2], OK
Menu:   Output, Field Output Requests, Edit, F-Output-1
    Output Variables: Preselected Defaults, # add: [SDV], OK
```

vi. Adding loads and BC

```
Module: Load
Menu:   BC, Manager
    Create, Name [BC-1], Step: Initial, Symm/Anti/Enca, Cont
    # pick: node @ origin, Done, ENCASTRE, OK
    Create, Name [BC-2], Step: Initial, Symm/Anti/Enca, Cont
    # pick: line @ Y=0, Done, YSYMM, OK
    # transverse load imposed as displacement to create epsilon = 2%
    Create, Name [traction], Step: transverse, Disp/Rota, Cont
    # pick: line @ Y=10, Done, # checkmark: U2 [0.2], OK
    # close Boundary Condition Manager pop-up window
```

vii. *Meshing the model. Use only one element because you will have uniform stress/strain.*

```
Module: Mesh
Menu:   Seed, Instance, Approximate global size [10], OK
Menu:   Mesh, Element Type
    Geometric Order: Quadratic # CPS8R avoid hourglass, OK
Menu:   Mesh, Instance, Yes
```

viii. *Creating a set to track the stresses and strains in the element*

```
Module: Mesh
Menu:   Tools, Set, Create
    Type: Element, Cont, # pick: the only element in the mesh, Done
Module: Step
Menu:   Output, History Output Requests, Edit, H-Output-1
    Domain: Set: Set-1, # expand: Stresses, # checkmark: S
    # expand: Strains, # checkmark: E, OK
```

ix. *Solving and visualizing the results*

```
Module: Job
Menu:   Job, Manager
    Create, Cont
    Tab: General, User subroutine file [umatps85.for], OK, OK
    Submit, # when Completed, Results
Module: Visualization
Menu:   Result, History Output
    # select: E22 at Element 1 Int Point 1, Save As [e22], OK
    # select: S22 at Element 1 Int Point 1, Save As [s22], OK
    # close History Output pop-up window
Menu:   Tools, XYData, Create
    Source: Operate on XY data, Cont, Operations: Combine
    # select: e22, Add to Expr., # select: s22, Add to Expr.
    Save As [s22-vs-e22], OK, Plot Expression
```

A shear load can be applied easily modifying the current `cae` *model.*

i. *Saving current* `cae` *model and creating a new one*

```
Menu:   File, Save
Menu:   File, Save As, [C:\SIMULIA\User\Ex_8.5\Ex_8.5.she.cae], OK
```

ii. *Defining analysis step*

```
Module: Step
Menu:   Step, Manager
    # select: transverse, Rename, [shear], OK
    Edit, Tab: Incrementation, Type: Fixed
    Maximum number of increments [50], Increment size [3], OK
    # close Step Manager pop-up window
```

iii. *Adding BC and loads*

```
Module: Load
Menu:   BC, Manager
    # shear loading imposed as displacement U1 on step: shear
    # select: traction, Rename, [shear], OK
    Edit, # uncheckmark: U2, # checkmark: U1 [0.4], OK # gamma_6=4%
    # close BC Manager pop-up window
```

iv. *Adding constraint to guarantee pure shear loading*

```
Module: Interaction
Menu:   Constraint, Create
    Type: Tie, Cont
    Surface, # pick: line @ x=0, Done
    Surface, # pick: line @ x=10, Done
    Discretization method: Surface to surface
    Position Tolerance: Specify distance [10]
    # uncheckmark: Adjust slave surface initial position, OK
```

v. *Solving and visualizing the results*

```
Module: Job
Menu:   Job, Manager
    Submit, # when Completed, Results
Module: Visualization
Menu:   Result, History Output
    # select: E12 at Element 1 Int Point 1, Save As [e12], OK
    # select: S12 at Element 1 Int Point 1, Save As [s12], OK
    # close History Output pop-up window
Menu:   Tools, XYData, Create
    Source: Operate on XY data, Cont, Operations: Combine
    # select: e12, Add to Expr., # select: s12, Add to Expr.
    Save As [s12-vs-e12], OK, Plot Expression
```

The model response for the two cases is shown in Figure 8.9.

8.5 Damage and Plasticity

For polymer matrix composites reinforced by strong and stiff fibers, damage and its conjugate thermodynamic force can be described by second-order tensors **D** and **Y**. Furthermore, the hardening processes that take place during plasticity and damage imply additional *dissipation*, so that

$$\rho\pi = T\rho\pi_s = \boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}}^p + R\,\dot{p} + \mathbf{Y} : \dot{\mathbf{D}} + \gamma\dot{\delta} \tag{8.122}$$

where (R, p) is the thermodynamic force-flux pair associated to plastic hardening, and (γ, δ) is the thermodynamic force-flux pair associated to damage hardening, and $\rho\pi$ is the dissipation heat due to irreversible phenomena.

For the particular case of (8.122), from (8.92) and (8.99), the following definitions for the thermodynamic forces are obtained

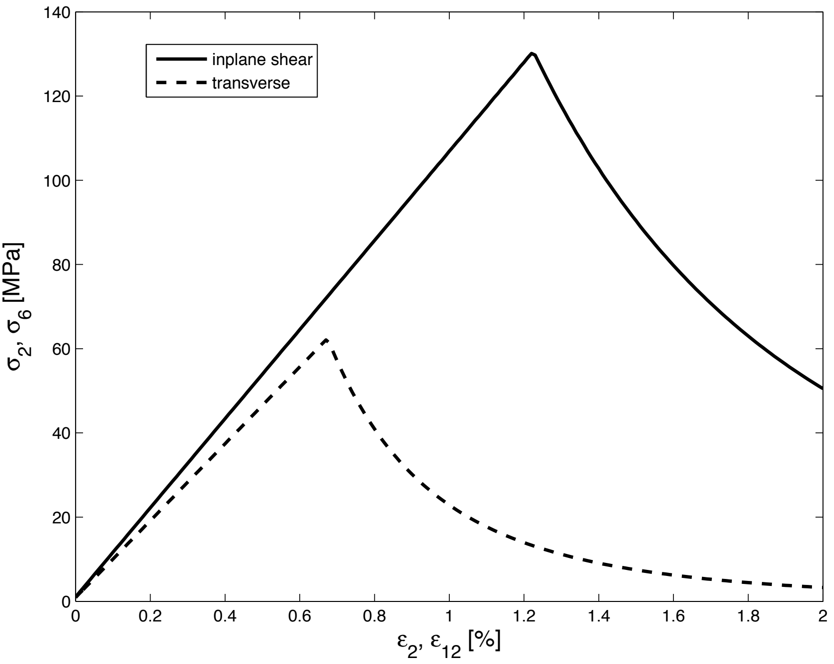


Fig. 8.9: Response to in-plane shear stress and to transverse tensile stress.

$$\boldsymbol{\sigma} = \rho \frac{\partial \psi}{\partial \boldsymbol{\varepsilon}} = -\rho \frac{\partial \psi}{\partial \boldsymbol{\varepsilon}^p} \quad \boldsymbol{\varepsilon} = \rho \frac{\partial \chi}{\partial \boldsymbol{\sigma}} \quad \mathbf{Y} = -\rho \frac{\partial \psi}{\partial \mathbf{D}} = \rho \frac{\partial \chi}{\partial \mathbf{D}} \tag{8.123}$$

as well as definitions for the hardening equations

$$\gamma = \rho \frac{\partial \chi}{\partial \delta} = -\rho \frac{\partial \psi}{\partial \delta} = \rho \frac{\partial \pi}{\partial \delta} \qquad R = \rho \frac{\partial \chi}{\partial p} = -\rho \frac{\partial \psi}{\partial p} = \rho \frac{\partial \pi}{\partial p} \tag{8.124}$$

The additive decomposition [63]

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^e + \boldsymbol{\varepsilon}^p \tag{8.125}$$

can be rewritten taking into account that the elastic component of strain can be calculated from stress and compliance, so that

$$\boldsymbol{\varepsilon} = \mathbf{S} : \boldsymbol{\sigma} + \boldsymbol{\varepsilon}^p \tag{8.126}$$

Therefore, the strain-stress law in incremental and rate form is

$$\begin{aligned} \delta \boldsymbol{\varepsilon} &= \mathbf{S} : \delta \boldsymbol{\sigma} + \delta \mathbf{S} : \boldsymbol{\sigma} + \delta \boldsymbol{\varepsilon}^p \\ \dot{\boldsymbol{\varepsilon}} &= \mathbf{S} : \dot{\boldsymbol{\sigma}} + \dot{\mathbf{S}} : \boldsymbol{\sigma} + \dot{\boldsymbol{\varepsilon}^p} \end{aligned} \tag{8.127}$$

showing that an increment of strain has three contributions, elastic, damage, and plastic. The elastic strain occurs as a direct result of an increment in stress, the damage strain is caused by the increment in compliance as the material damages, and the plastic strain occurs at constant compliance. The elastic unloading stiffness does not change due to plasticity but it reduces due to damage. Following this argument, it is customary [92] to assume that the free energy and complementary free energy can be separated as follows

$$\begin{aligned}\psi(\boldsymbol{\varepsilon},\boldsymbol{\varepsilon}^p,p,\mathbf{D},\delta)&=\psi^e(\boldsymbol{\varepsilon}^e,\mathbf{D},\delta)+\psi^p(\boldsymbol{\varepsilon}^p,p)\\ \chi(\boldsymbol{\sigma},\boldsymbol{\varepsilon}^p,p,\mathbf{D},\delta)&=\chi^e(\boldsymbol{\sigma},\mathbf{D},\delta)+\chi^p(\boldsymbol{\varepsilon}^p,p)\end{aligned} \tag{8.128}$$

Suggested Problems

Problem 8.1 *Using the formulation and properties of Example 8.2 (p. 292), obtain a graphical representation of the evolution strain vs. nominal stress (ε vs. σ) and the evolution of strain vs. effective stress (ε vs. $\widetilde{\sigma}$) for a point on the top surface of the beam and for another point on the bottom surface of the beam. Comment on the graphs obtained.*

Problem 8.2 *Implement a UMAT for a one-dimensional CDM active in the x_1-direction only. Use 2D plane stress constitutive equations. Leave the x_2-direction, Poisson's, and shear terms as linear elastic with no damage. Verify the program by recomputing Example 8.2 and the plots obtained in Problem 8.1. Note that to obtain the same values, the Poisson's ratio should be set to zero.*

Problem 8.3 *The Gibbs free energy is defined in expanded form and using Voigt contracted notation as*

$$\chi=\frac{1}{2\,\rho}\quad\left[\frac{\sigma_1^2}{(1-D_1)^2\,\widetilde{E}_1}+\frac{\sigma_2^2}{(1-D_2)^2\,\widetilde{E}_2}+\frac{\sigma_6^2}{(1-D_1)\,(1-D_2)\,\widetilde{G}_{12}}-\right.$$
$$\left.-\left(\frac{\widetilde{\nu}_{21}}{\widetilde{E}_2}+\frac{\widetilde{\nu}_{12}}{\widetilde{E}_1}\right)\frac{\sigma_1\sigma_2}{(1-D_1)\,(1-D_2)}\right]$$

where $\widetilde{E}_1$, $\widetilde{E}_2$, $\widetilde{\nu}_{12}$, $\widetilde{\nu}_{21}$, and $\widetilde{G}_{12}$ are the undamaged in-plane elastic orthotropic properties of a unidirectional lamina where the subindex $()_1$ denotes the fiber direction and $()_2$ denotes the transverse direction. (a) Obtain the secant constitutive equations, $\mathbf{C}$ and $\mathbf{S}$, using the given Gibbs free energy. (b) Obtain the thermodynamic forces Y_1 and Y_2 associated to D_1 and D_2. (c) If $\mathbf{M}$ is represented using Voigt contracted notation and multiplied by a Reuter matrix as

$$\mathbf{M}=\begin{bmatrix}(1-D_1)&0&0\\0&(1-D_2)&0\\0&0&\sqrt{1-D_1}\sqrt{1-D_2}\end{bmatrix}$$

check if this definition of $\mathbf{M}$ can be used as the damage effect tensor in a damage model using the principle of energy equivalence. Justify and comment on your conclusion.

Problem 8.4 *The damage activation function, for the model shown in Problem 8.3, is defined as*

Table 8.3: Elastic properties for composite material lamina.

$\widetilde{E}_1$	$\widetilde{E}_2$	$\widetilde{G}_{12}$	$\widetilde{\nu}_{12}$
171.4 GPa	9.08 GPa	5.29 GPa	0.32

Table 8.4: Identification model parameters for composite material lamina.

H_1	H_2	γ_0	c_1	c_2
0.024	8.36	1.0	1.5	-2.8

$$g := \hat{g} - \hat{\gamma} = \sqrt{{Y_1}^2 H_1 + {Y_2}^2 H_2} - (\gamma + \gamma_0)$$

where H_1 and H_2 are model parameters that depend on elastic and strength material properties, and Y_1 and Y_2 are the thermodynamic forces associated to the damage variables D_1 and D_2, respectively. The damage hardening depends on δ according to

$$\hat{\gamma} = \gamma + \gamma_0 = c_1 \left[\exp\left(\frac{\delta}{c_2}\right) - 1 \right] + \gamma_0$$

where γ_0 defines the initial threshold value, and c_1 and c_2 are material parameters. All necessary material parameters are shown in Tables 8.3 and 8.4.

a) Using a flow chart diagram, describe the algorithm, with all necessary steps to implement it as a constitutive subroutine in a finite element package.

b) Compute the analytic expressions necessary to implement the model in a UMAT.

c) Program the algorithm using the UMAT capability for a plane stress constitutive equation.

d) Finally, using Abaqus*, plot a single curve of apparent stress σ_2 vs. apparent strain ε_2 for a RVE loaded only with ε_2.*

e) Using pseudo code, describe the process used to solve the problem in Abaqus/CAE*.*

Chapter 9

Discrete Damage Mechanics

Prediction of damage initiation and propagation is tackled in Chapter 8 using continuum damage mechanics (CDM). Alternatives to CDM include: micromechanics of damage, crack opening displacement methods, computational micromechanics, and synergistic methods. While CDM (Chapter 8) homogenizes the damage and treats it phenomenologically, the alternative methods attempt to represent the actual geometry and characteristics of damage.

Prediction of transverse matrix cracking in laminated composites has been extensively studied for the particular case of symmetric $[0_m/90_n]_S$ laminates under membrane loads, for which matrix cracking is found in the 90° laminas (transverse laminas). Extensions to other laminate configurations, such as $[0/\pm\theta/0]_S$ and $[0/\theta_1/\theta_2]_S$, featuring cracks in the off-axis θ laminas, have been developed, but they are still limited to symmetric laminates subjected to in-plane loading.

Micromechanics of Damage Models (MMD) find an approximate elasticity solution for a laminate with a discrete crack or cracks [93–114]. The solutions are approximate because kinematic assumptions are made, such as a linear [115] or bilinear [116] distribution of interlaminar shear stress through the thickness of each lamina, as well as particular spatial distributions of in-plane displacement functions [113], stresses, and so on. The state variable is the crack density in the cracking lamina, defined as the number of cracks per unit distance perpendicular to the crack surface. Therefore, the state variable is measurable. One advantage of MMD is that the reduction of laminate moduli as a function of crack density is calculated without resorting to additional parameters as in the case of CDM. The main disadvantage of MMD is that most of the solutions available are limited to symmetric laminates under membrane loads with only one or two laminas cracking. A generalization to the case of multiple cracking laminas is presented in this chapter by resorting to the concept of synergistic methods, explained below.

Crack Opening Displacement (COD) methods [117–124] are based on the theory of elastic bodies with voids [125]. The distinct advantage of COD models is that the laminate stiffness can be calculated for any laminate configuration, even non-symmetric laminate stacking sequence (LSS), subject to any deformation, including bending, featuring matrix cracking in any of its laminas [126].

Numerical solutions, such as FEA, provide 3D solutions without the kinematic

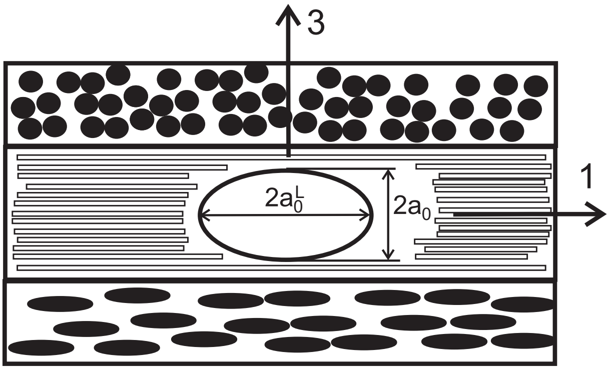


Fig. 9.1: Representative crack geometry.

simplifications of MMD and COD models [104, 120, 123, 127–130]. However, FEA solutions require a new mesh and boundary conditions for each LSS, crack orientation, and so on, making them too cumbersome for practical application. Another numerical approach is Monte Carlo simulation, where the probabilistic distribution of flaws in material is considered [131–133]. Unfortunately, Monte Carlo simulations require additional parameters that have to be adjusted by fitting the results of the model to experimental damage evolution data. Such data are scarce.

Synergistic Damage Mechanics (SDM) methods combine elements of different modeling strategies such as CDM and MMD [131, 132, 134–138], bringing the best features of each of the models involved. For example, in this chapter the laminate stiffness reduction is computed via MMD methods and the generalization to multiple cracking laminas is made via CDM concepts, but unlike CDM models, no additional parameters are needed.

9.1 Overview

In the following we describe how to use two material properties, the fracture toughness in modes I and II, G_{Ic}, G_{IIc}, to predict the damaging behavior and transverse tensile failure of a unidirectional fiber reinforced lamina embedded in a laminate. The constraining effect of adjacent laminas is taken into account, leading to apparent transverse tensile strength F_{2t} being a function of ply thickness. The crack initiation strain, crack density evolution as a function of stress (strain) up to crack saturation, and stress redistribution to adjacent laminas are predicted accurately.

The physics of matrix cracking under transverse tension and in-plane shear is as follows. No matter how much care is taken during the production process, there are always defects in the material. These defects may be voids, microcracks, fiber–matrix debonding, and so on, but all of them can be represented by a typical matrix crack of representative size $2a_0$, as shown in Figure 9.1.

When subject to load, matrix cracks grow parallel to the fiber orientation, as shown in Figure 9.2, where it can be seen that cracks are aligned with the fiber

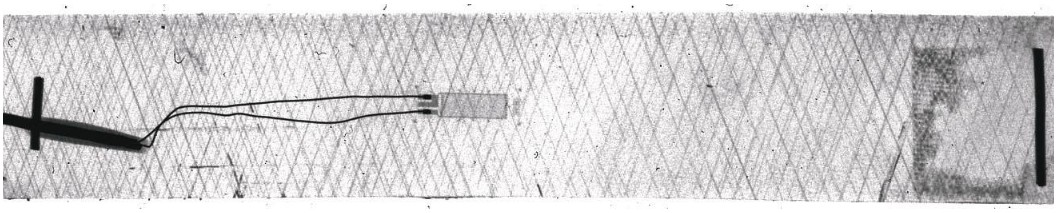


Fig. 9.2: Matrix cracks in the ±70 laminas of a $[0/\pm 70_4/0_{1/2}]_S$ laminate loaded to 0.7% tensile strain along the 0° direction [139].

direction in the ±70° laminas. These sets of parallel cracks reduce the stiffness of the cracked lamina, which then sheds its share of the load onto the remaining laminas. In each lamina, the damage caused by this set of parallel cracks is represented by the crack density, defined as the inverse of the distance between two adjacent cracks $\lambda = 1/(2l)$, as shown in Figure 9.3. Therefore, the crack density is the only state variable needed to represent the state of damage in the cracked lamina. Note that the actual, discrete cracks are modeled by the theory, which is thus named *discrete damage mechanics* (DDM).

The basic ingredients of the DDM model for transverse tension and in-plane shear damage are listed below:

i. In each lamina i, the state variable is the crack density λ_i. Two damage variables $D_2(\lambda_i)$ and $D_6(\lambda_i)$ are defined for convenience but they are not independent variables; instead, they are computed in terms of the crack density. The set of crack densities for the laminate is denoted by $\lambda = \lambda_i$ with $i = 1...N$, where N is the number of laminas in the laminate.

ii. The independent variable is the midsurface[1] strain $\epsilon = \{\epsilon_1, \epsilon_2, \gamma_{12}\}^T$.

iii. The damage activation function, which separates the damaging states from the undamaging states, is written as follows

$$g = \max\left[\frac{G_I(\lambda,\epsilon,\Delta T)}{G_{Ic}}, \frac{G_{II}(\lambda,\epsilon,\Delta T)}{G_{IIc}}\right] - 1 \leq 0 \tag{9.1}$$

where $g \leq 0$ represents the undamaging domain. The critical ERRs are not easily found in the literature but they can be fit to available experimental data using the methodology explained, for example in [55, 140].

iv. The damage threshold is embedded into g, and represented by the (invariant) material properties G_{Ic}, G_{IIc}. Before damage starts, $\lambda = 0$ and (9.1) is a damage initiation criterion, similar to [141] but without mode interaction. With $\lambda = 0$, the strain for which $g = 0$ is the strain for crack initiation. Once damage starts, (9.1) becomes a damage activation function by virtue of the automatic hardening described below.

[1] The analysis presented in this section is for symmetric laminates under membrane forces. A formulation for unsymmetric laminates and/or laminates under bending is being reported in [126].

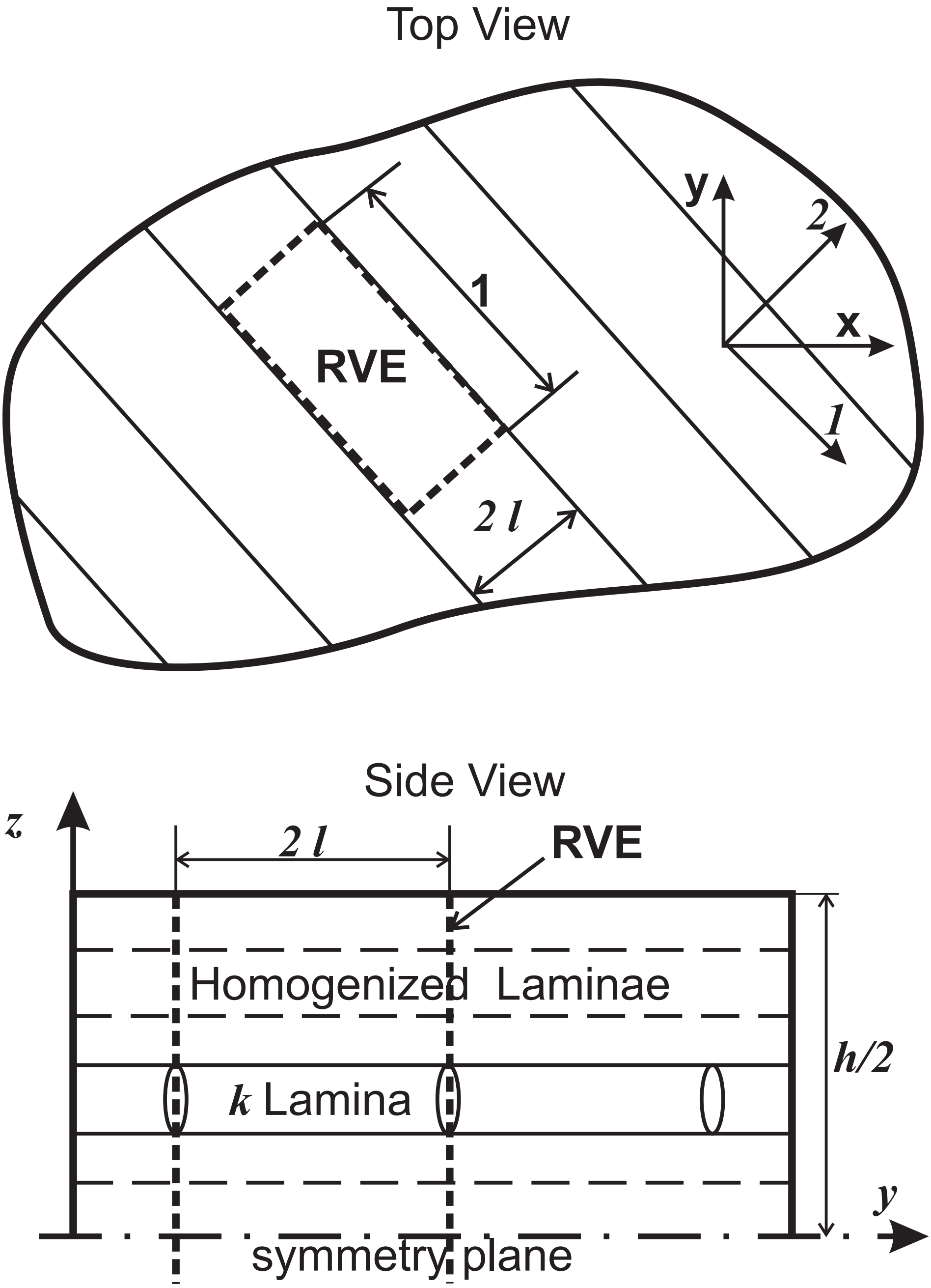


Fig. 9.3: Representative unit cell used in discrete damage mechanics.

v. The hardening function is embedded into the damage activation function g. For a given value of strain, the calculated values of energy release rate $G_I(\lambda), G_{II}(\lambda)$ are monotonically decreasing functions of λ. Therefore, as soon as λ grows, $G_I(\lambda), G_{II}(\lambda)$ decrease, making $g < 0$ and thus stopping further damage until the *driving thermodynamic force*, i.e., the strain, is increased by the application of additional load.

vi. No damage evolution function need to be postulated, with the advantage that no new empirical parameters are needed. Simply the crack density λ adjusts itself to a value that will set the laminate in equilibrium with the external loads for the current strain while satisfying $g = 0$. A return mapping algorithm (Section 8.4.1) achieves this by iterating until $g = 0$ and updating the crack density with iterative increments calculated as $\Delta\lambda = -g/\frac{\partial g}{\partial \lambda}$.

vii. The crack density grows until the lamina is saturated with cracks ($\lambda \to \infty$). At that point the lamina loses all of its transverse and shear stiffness ($D_2 \approx 1, D_6 \approx 1$), at which point all of the load is already transferred to the remaining laminas in the laminate. The analysis of the cracked lamina is stopped when the crack density reaches $\lambda_{lim} = 1/h_k$, where h_k is the thickness of lamina k; i.e., when cracks are closely spaced at a distance equal to the lamina thickness.

Having described the ingredients of the model, it now remains to show how to calculate the various quantities. The solution begins by calculating the degraded[2] stiffness of the laminate $Q = [A]/h$ for a given crack density λ_k in a cracked lamina k, where $[A]$ is the in-plane laminate stiffness matrix,[3] and h is the thickness of the laminate.

The following conventions are used in this section:

- (i) denotes any lamina in the laminate.
- (k) denotes the cracking lamina.
- (m) denotes any lamina other than the cracking one ($m \neq k$).
- A sub/superscript in parenthesis (i) denotes the lamina number; not a power or order of differentiation.
- x_j with $j = 1, 2, 3$ denote the coordinates x_1, x_2, x_3, or $j = 1, 2, 6$ for quantities expressed in Voigt contracted notation.
- $u(x_j), v(x_j), w(x_j)$ with $j = 1, 2, 3$ are the three components of the displacement.
- *hat* $\hat{p}$ denotes the thickness average of quantity p, where the thickness is mentioned or it is obvious from context.
- *tilde* $\widetilde{p}$ denotes the virgin value of quantity p.
- *over-line* $\overline{p}$ denotes the volume average of quantity p.

[2] Also called “damaged”, “reduced”, or “homogenized”.

[3] Not to be confused with the crack area A.

9.2 Approximations

Most practical laminates are symmetric and the most efficient use of them is by designing the structure to be loaded predominantly with membrane loads [1, Chapter 12]. Therefore, the solution presented here is for a symmetric laminate under membrane loads. In this case,

$$\frac{\partial w^{(i)}}{\partial x_1} = \frac{\partial w^{(i)}}{\partial x_2} = 0 \tag{9.2}$$

where $u(x_j), v(x_j), w(x_j)$ with $j = 1, 2, 3$ are the displacements of a point in lamina i as a function of the coordinates x_j with $j = 1, 2, 3$. Furthermore, the thickness h_i of the laminas are assumed to be small, so that the plane stress assumption holds

$$\sigma_3^{(i)} = 0 \tag{9.3}$$

Since all cracks are parallel to the fiber direction and practical designs avoid thick laminas, it can be expected that the cracks occupy the entire thickness of the lamina. Any crack smaller than the lamina thickness is unstable both through the lamina thickness and along the fiber direction [1, Section 7.2.1].

Since the objective is to calculate the laminate stiffness reduction due to cracks, it suffices to work with thickness averages of the variables. A thickness average is denoted by

$$\hat{\phi} = \frac{1}{h'} \int_{h'} \phi \, dx_3 \quad ; \quad h' = \int dx_3 \tag{9.4}$$

where h' can be the lamina or laminate thickness, denoted by h_i, h, respectively.

Specifically,

- $\hat{u}^{(i)}(x_j), \hat{v}^{(i)}(x_j), \hat{w}^{(i)}(x_j)$ are the thickness-average displacements in lamina i as a function of the in-plane coordinates x_j with $j = 1, 2$.
- $\hat{\epsilon}_1^{(i)}(x_j), \hat{\epsilon}_2^{(i)}(x_j), \hat{\gamma}_{12}^{(i)}(x_j)$ are the thickness-average strains in lamina i.
- $\hat{\sigma}_1^{(i)}(x_j), \hat{\sigma}_2^{(i)}(x_j), \hat{\tau}_{12}^{(i)}(x_j)$ are the thickness-average stress in lamina i.

Out-of-plane (intralaminar) shear stress components appear due to the perturbation of the displacement field caused by the crack. These are approximated by linear functions through the thickness of the lamina i, as follows

$$\begin{aligned}
\tau_{13}^{(i)}(x_3) &= \tau_{13}^{i-1,i} + \left(\tau_{13}^{i,i+1} - \tau_{13}^{i-1,i}\right) \frac{x_3 - x_3^{i-1,i}}{h_i} \\
\tau_{23}^{(i)}(x_3) &= \tau_{23}^{i-1,i} + \left(\tau_{23}^{i,i+1} - \tau_{23}^{i-1,i}\right) \frac{x_3 - x_3^{i-1,i}}{h_i}
\end{aligned} \tag{9.5}$$

where x_3^{i-1} is the thickness coordinate at the bottom of lamina i, i.e., at the interface between lamina $i-1$ and lamina i, and $\tau_{13}^{i-1,i}$ is the shear stress at the interface between the $i-1$ and the i lamina. This assumption, which is common to several other analytical models, is called the *shear lag assumption*. The linear approximation has been shown to yield accurate results [130].

The shear lag equations are obtained from the constitutive equations for out-of-plane shear strains and stresses using weighted averages [136, Appendix A],

$$
\begin{aligned}
\left\{ \begin{array}{c} \hat{u}^{(i)} - \hat{u}^{(i-1)} \\ \hat{v}^{(i)} - \hat{v}^{(i-1)} \end{array} \right\} &= \tfrac{h_{(i-1)}}{6} \left[\begin{array}{cc} S_{45} & S_{55} \\ S_{44} & S_{45} \end{array} \right]^{(i-1)} \left\{ \begin{array}{c} \tau_{23}^{i-2,i-1} \\ \tau_{13}^{i-2,i-1} \end{array} \right\} \\
&+ \left[\tfrac{h_{(i-1)}}{3} \left[\begin{array}{cc} S_{45} & S_{55} \\ S_{44} & S_{45} \end{array} \right]^{(i-1)} + \tfrac{h_{(i)}}{3} \left[\begin{array}{cc} S_{45} & S_{55} \\ S_{44} & S_{45} \end{array} \right]^{(i)} \right] \left\{ \begin{array}{c} \tau_{23}^{i-1,i} \\ \tau_{13}^{i-1,i} \end{array} \right\} \\
&+ \tfrac{h_{(i)}}{6} \left[\begin{array}{cc} S_{45} & S_{55} \\ S_{44} & S_{45} \end{array} \right]^{(i)} \left\{ \begin{array}{c} \tau_{23}^{i,i+1} \\ \tau_{13}^{i,i+1} \end{array} \right\} \qquad (9.6)
\end{aligned}
$$

Inverting (9.6), the intralaminar stresses are written in terms of displacements at the interfaces, as follows

$$
\begin{aligned}
\tau_{23}^{i,i+1} - \tau_{23}^{i-1,i} &= \sum_{j=1}^{n-1} \left[[H]^{-1}{}_{2i-1,2j-1} - [H]^{-1}{}_{2i-3,2j-1} \right] \left\{ \hat{u}^{(j+1)} - \hat{u}^{(j)} \right\} \\
&+ \left[[H]^{-1}{}_{2i-1,2j} - [H]^{-1}{}_{2i-3,2j} \right] \left\{ \hat{v}^{(j+1)} - \hat{v}^{(j)} \right\} \\
\tau_{13}^{i,i+1} - \tau_{13}^{i-1,i} &= \sum_{j=1}^{n-1} \left[[H]^{-1}{}_{2i,2j-1} - [H]^{-1}{}_{2i-2,2j-1} \right] \left\{ \hat{u}^{(j+1)} - \hat{u}^{(j)} \right\} \\
&+ \left[[H]^{-1}{}_{2i,2j} - [H]^{-1}{}_{2i-2,2j} \right] \left\{ \hat{v}^{(j+1)} - \hat{v}^{(j)} \right\} \qquad (9.7)
\end{aligned}
$$

in terms of the $2(N-1)$ by $2(N-1)$ coefficient matrix H.

9.3 Lamina Constitutive Equation

The stress-strain law for the cracking lamina k is that of an intact material, i.e.,

$$
\hat{\sigma}_i^{(k)} = \widetilde{Q}_{ij}^{(k)} \left(\hat{\epsilon}_j^{(k)} - \alpha_j^{(k)} \Delta T \right) \qquad (9.8)
$$

where $\alpha^{(k)}$ is the CTE of lamina k, $\sigma_i^{(k)} = \left\{ \sigma_1^{(k)}, \sigma_2^{(k)}, \tau_{12}^{(k)} \right\}^T$, and *tilde* denotes a virgin property. The strain-displacement equations are

$$
\epsilon^{(k)} = \left\{ \begin{array}{l} \epsilon_1^{(k)} = u_{,1}^{(k)} \\ \epsilon_2^{(k)} = v_{,2}^{(k)} \\ \gamma_{12}^{(k)} = u_{,2}^{(k)} + v_{,1}^{(k)} \end{array} \right\} \qquad (9.9)
$$

For the remaining laminas ($m \neq k$), the constitutive equations can be obtained using (9.8) and the stiffness matrix $Q_{ij}^{(m)}$, written in terms of their previously calculated damage values $D_2^{(m)}, D_6^{(m)}$, defined in (9.32), and rotated to the k coordinate system using the usual transformation equations [1, Section 5.4]

$$Q^{(m)} = T^{-1} \begin{bmatrix} \widetilde{Q}_{11}^{(m)} & \left(1 - D_2^{(m)}\right) \widetilde{Q}_{12}^{(m)} & 0 \\ \left(1 - D_2^{(m)}\right) \widetilde{Q}_{12}^{(m)} & \left(1 - D_2^{(m)}\right) \widetilde{Q}_{22}^{(m)} & 0 \\ 0 & 0 & \left(1 - D_6^{(m)}\right) \widetilde{Q}_{66}^{(m)} \end{bmatrix} T^{-T} \tag{9.10}$$

where $T^{-1} = [T(-\theta)]$ and $T^{-T} = [T(-\theta)]^T$; see [1, (5.35)].

9.4 Displacement Field

The objective now is to solve for the average displacements $\hat{u}^{(i)}(x_j), \hat{v}^{(i)}(x_j); j = 1, 2,$ in all laminas i for a given crack density set λ and applied strain ϵ. Taking into account that the intralaminar shear stresses are assumed to vary linearly through the thickness of each lamina, the equilibrium equations (1.15) for each lamina can be written as follows

$$\hat{\sigma}_{1,1}^{(i)} + \hat{\tau}_{12,2}^{(i)} + \left(\hat{\tau}_{13}^{i,i+1} - \hat{\tau}_{13}^{i-1,i}\right)/h_i = 0 \tag{9.11}$$

$$\hat{\tau}_{12,1}^{(i)} + \hat{\sigma}_{2,2}^{(i)} + \left(\hat{\tau}_{23}^{i,i+1} - \hat{\tau}_{23}^{i-1,i}\right)/h_i = 0 \tag{9.12}$$

Using the strain-displacement equations (9.9) and the constitutive equations (9.8) into the equilibrium equations (9.11)-(9.12) leads to a system of $2N$ partial differential equations (PDE) in $\hat{u}^{(i)}(x_j), \hat{v}^{(i)}(x_j)$. The PDE has particular solutions of the form

$$\begin{aligned} \hat{u}^{(i)} &= a_i \sinh \lambda_e x_2 + a\ x_1 + b\ x_2 \\ \hat{v}^{(i)} &= b_i \sinh \lambda_e x_2 + b\ x_1 + a^* x_2 \end{aligned} \tag{9.13}$$

where e is the eigenvalue number. The general solution can be written as

$$\begin{Bmatrix} \hat{u}^{(1)} \\ \hat{u}^{(2)} \\ . \\ . \\ . \\ \hat{u}^{(n)} \\ \hat{v}^{(1)} \\ \hat{v}^{(2)} \\ . \\ . \\ . \\ \hat{v}^{(n)} \end{Bmatrix} = \sum_{e=1}^{2N} A_e \begin{Bmatrix} a_1 \\ a_2 \\ . \\ . \\ . \\ a_n \\ b_1 \\ b_2 \\ . \\ . \\ . \\ b_n \end{Bmatrix}_e sinh\left(\eta_e x_2\right) + \begin{Bmatrix} a \\ a \\ . \\ . \\ . \\ a \\ b \\ b \\ . \\ . \\ . \\ b \end{Bmatrix} x_1 + \begin{Bmatrix} b \\ b \\ . \\ . \\ . \\ b \\ a^* \\ a^* \\ . \\ . \\ . \\ a^* \end{Bmatrix} x_2 \tag{9.14}$$

which substituted into the PDE leads to the eigenvalue problem

$$\begin{bmatrix} \alpha_1 & \beta_1 \\ \alpha_2 & \beta_2 \end{bmatrix} \begin{Bmatrix} a_j \\ b_j \end{Bmatrix} + \eta^2 \begin{bmatrix} \zeta_{26} & \zeta_{22} \\ \zeta_{66} & \zeta_{26} \end{bmatrix} \begin{Bmatrix} a_j \\ b_j \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \tag{9.15}$$

where $j = 1...2N$; η are the $2N$ eigenvalues and $\{a_j, b_j\}^T$ are the $2N$ eigenvectors of (9.15).

It turns out that two of the eigenvalues are always zero (corresponding to the linear terms in (9.14)), which can be taken to be the last two in the set, thus remaining only $2N - 2$ independent solutions. Then, the general solution of the PDE system is built as the linear combination of the $2N - 2$ independent solutions as follows

$$\begin{Bmatrix} \hat{u}^{(i)} \\ \hat{v}^{(i)} \end{Bmatrix} = \sum_{e=1}^{2N-2} A_e \begin{Bmatrix} a_i \\ b_i \end{Bmatrix}_e \sinh(\eta_e x_2) + \begin{Bmatrix} a \\ b \end{Bmatrix} x_1 + \begin{Bmatrix} b \\ a^* \end{Bmatrix} x_2 \tag{9.16}$$

where A_e are unknown coefficients in the linear combination. It can be seen that the general solution contains $2N + 1$ unknown coefficients, including the scalars a, b, a^* and the sets A_e with $e = 1...2N - 2$. To determine these coefficients, one needs $2N + 1$ boundary conditions on the boundary of the representative volume element (RVE) in Figure 9.3. Note that the RVE spans a unit length along the fiber direction x_1, a distance $2l$ between successive cracks (along x_2) and the whole thickness h of the symmetric laminate.

Two very important parameters are introduced through the boundary conditions, namely the crack density λ and the stress $\hat{\sigma} = \{N\}/h$ applied to the laminate, where $\{N\}$ represents the three components of the in-plane force per unit length.[4] The crack density enters through the dimension of the RVE, which has a width of $2l = 1/\lambda$. The applied stress (or strain) enters through the force equilibrium on the RVE. In summary, there are $2N + 1$ boundary conditions that lead to a system of $2N + 1$ algebraic equations that can be solved for the $2N + 1$ coefficients in (9.16). Therefore, the average displacements in all laminas are now known from (9.16) for given values of crack density λ and applied load $\hat{\sigma} = \{N\}/h$.

9.4.1 Boundary Conditions for $\Delta T = 0$

First consider the case of mechanical loads and no thermal loads. To find the values of $A_e, a, a*, b$, the following boundary conditions are enforced: (a) stress-free at the crack surfaces, (b) external loads, and (c) homogeneous displacements. The boundary conditions are then assembled into an algebraic system as follows

$$[B] \left\{A_e, a, a^*, b\right\}^T = \{F\} \tag{9.17}$$

where $[B]$ is the coefficient matrix of dimensions $2N + 1$ by $2N + 1$; $\left\{A_e, a, a^*, b\right\}^T$ represents the $2N + 1$ unknown coefficients, and $\{F\}$ is the RHS or force vector, also of dimension $2N + 1$.

[4] Not to be confused with the number of laminas N.

(a) Stress-free at the Crack Surfaces

The surfaces of the cracks are stress-free

$$\int_{-1/2}^{1/2} \hat{\sigma}_2^{(k)}\left(x_1, l\right) \mathrm{d}x_1 = 0 \tag{9.18}$$

$$\int_{-1/2}^{1/2} \hat{\tau}_{12}^{(k)}\left(x_1, l\right) \mathrm{d}x_1 = 0 \tag{9.19}$$

(b) External Loads

In the direction parallel to the surface of the cracks (fiber direction x_1) the load is supported by all the laminas

$$\frac{1}{2l}\sum_{i=1}^{N} h_i \int_{-l}^{l} \hat{\sigma}_1^{(i)}(1/2, x_2) dx_2 = h\hat{\sigma}_1 \tag{9.20}$$

In the direction normal to the crack surface (x_2 direction) only the uncracking (homogenized) laminas carry load

$$\sum_{m \neq k} h_m \int_{1/2}^{1/2} \hat{\sigma}_2^{(m)}\left(x_1, l\right) dx_1 = h\hat{\sigma}_2 \tag{9.21}$$

$$\sum_{m \neq k} h_m \int_{1/2}^{1/2} \hat{\tau}_{12}^{(m)}(x_1, l) dx_1 = h\hat{\tau}_{12} \tag{9.22}$$

(c) Homogeneous Displacements

For a homogenized symmetric laminate, membrane loads produce a uniform displacement field through the thickness, i.e., all the uncracking laminas are subjected to the same displacement

$$\hat{u}^{(m)}\left(x_1, l\right) = \hat{u}^{(r)}\left(x_1, l\right) \quad ; \quad \forall m \neq k \tag{9.23}$$

$$\hat{v}^{(m)}\left(x_1, l\right) = \hat{v}^{(r)}\left(x_1, l\right) \quad ; \quad \forall m \neq k \tag{9.24}$$

where r is an uncracked lamina taken as reference. In the computer implementation, lamina 1 is taken as reference unless lamina 1 is cracking, in which case lamina 2 is taken as reference.

9.4.2 Boundary Conditions for $\Delta T \neq 0$

Next, consider the case of thermal loads, which add a constant term to the boundary conditions. Constant terms do not affect the matrix $[B]$, but rather subtract from the forcing vector $\{F\}$, as follows

$$\{F\}_{\Delta T\neq 0} = \left\{ \begin{array}{c} \Delta T \sum\limits_{j=1,2,6} \bar{Q}_{1j}^{(k)} \bar{\alpha}_j^{(k)} \\ \Delta T \sum\limits_{j=1,2,6} \bar{Q}_{1j}^{(k)} \bar{\alpha}_j^{(k)} \\ \Delta T \sum\limits_{i\neq(k)} \sum\limits_{j=1,2,6} \bar{Q}_{1j}^{(i)} \bar{\alpha}_j^{(i)} \\ \Delta T \sum\limits_{i\neq k} \sum\limits_{j=1,2,6} \bar{Q}_{2j}^{(i)} \bar{\alpha}_j^{(i)} \\ \Delta T \sum\limits_{i\neq k} \sum\limits_{j=1,2,6} \bar{Q}_{6j}^{(i)} \bar{\alpha}_j^{(i)} \\ 0 \\ 0 \\ \ldots \\ \ldots \\ 0 \\ 0 \end{array} \right\} \tag{9.25}$$

In this way, the strain calculated for a unit thermal load ($\Delta T = 1$) is the degraded CTE of the laminate for the current crack density set λ.

9.5 Degraded Laminate Stiffness and CTE

In this section, we calculate the degraded stiffness of the laminate $Q = \mathbf{A}/h$ for a given crack density λ_k in a cracked lamina k, where $\mathbf{A}$ is the in-plane laminate stiffness matrix, and h is the thickness of the laminate. First, the thickness-averaged strain field in all laminas can be obtained by using the kinematic equations (9.9), namely by differentiating (9.16). Then, the compliance of the laminate S in the coordinate system of lamina k can be calculated one column at a time by solving for the strains (9.9) for three load cases, *a, b,* and *c,* all with $\Delta T = 0$, as follows

$${}^a\hat{\sigma} = \left\{ \begin{array}{c} 1 \\ 0 \\ 0 \end{array} \right\}; \quad {}^b\hat{\sigma} = \left\{ \begin{array}{c} 0 \\ 1 \\ 0 \end{array} \right\}; \quad {}^c\hat{\sigma} = \left\{ \begin{array}{c} 0 \\ 0 \\ 1 \end{array} \right\}; \quad \Delta\ T = 0 \tag{9.26}$$

Since the three applied stress states are unit values, for each case, *a, b, c,* the volume average of the strain (9.9) represents one column in the laminate compliance matrix

$$S = \begin{bmatrix} {}^a\epsilon_x & {}^b\epsilon_x & {}^c\epsilon_x \\ {}^a\epsilon_y & {}^b\epsilon_y & {}^c\epsilon_y \\ {}^a\gamma_{xy} & {}^b\gamma_{xy} & {}^c\gamma_{xy} \end{bmatrix} \tag{9.27}$$

where x, y are the coordinates of lamina k (Figure 9.3). Next, the laminate stiffness in the coordinate system of lamina k is

$$Q = S^{-1} \tag{9.28}$$

To get the degraded CTE of the laminate, one sets $\hat{\sigma} = \{0, 0, 0\}^T$ and $\Delta T = 1$. The resulting strain is equal to the CTE of the laminate, i.e., $\{\alpha_x, \alpha_y, \alpha_{xy}\}^T = \{\epsilon_x, \epsilon_y, \gamma_{xy}\}^T$.

9.6 Degraded Lamina Stiffness

The stiffness of lamina m, with $m \neq k$, in the coordinate system of lamina k (see Figure 9.3) is given by (9.10) in terms of the previously calculated values $D_2^{(m)}, D_6^{(m)}$, given by (9.32). The stiffness of the cracking lamina $Q^{(k)}$ is yet unknown. Note that all quantities are expressed in the coordinate system of lamina k.

The laminate stiffness is defined by the contribution of the cracking lamina k plus the contribution of the remaining $N-1$ laminas, as follows

$$Q = Q^{(k)} \frac{h_k}{h} + \sum_{m=1}^{n} (1 - \delta_{mk}) Q^{(m)} \frac{h_m}{h} \tag{9.29}$$

where the *delta* Dirac is defined as $\delta_{mk} = 1$ if $m = k$, otherwise 0. The left-hand side (LHS) of (9.29) is known from (9.28) and all values of $Q^{(m)}$ can be easily calculated since the m laminas are not cracking at the moment. Therefore, one can calculate the degraded stiffness $Q^{(k)}$ of lamina k as follows

$$Q^{(k)} = \frac{h}{h_k} \left[Q - \sum_{m=1}^{n} (1 - \delta_{mk}) Q^{(m)} \frac{h_m}{h} \right] \tag{9.30}$$

where Q without a superscript is the stiffness of the laminate.

To facilitate later calculations, the stiffness $Q^{(k)}$ can be written using concepts of continuum damage mechanics (Section 8.2) in terms of the stiffness of the undamaged lamina and damage variables $D_2^{(k)}, D_6^{(k)}$, as follows

$$Q^{(k)} = \begin{bmatrix} \widetilde{Q}_{11}^{(k)} & (1 - D_2)\, \widetilde{Q}_{12}^{(k)} & 0 \\ (1 - D_2)\, \widetilde{Q}_{12}^{(k)} & (1 - D_2)\, \widetilde{Q}_{22}^{(k)} & 0 \\ 0 & 0 & (1 - D_6)\, \widetilde{Q}_{66}^{(k)} \end{bmatrix} \tag{9.31}$$

with $D_j^{(k)}$ calculated for a given crack density λ_k and applied strain ϵ^0, as follows

$$D_j^{(k)}(\lambda_k, \epsilon^0) = 1 - Q_{jj}^{(k)} / \widetilde{Q}_{jj}^{(k)} \quad ; \quad j = 2, 6; \text{ no sum on } j \tag{9.32}$$

where $\widetilde{Q}^{(k)}$ is the original value of the undamaged property and $Q^{(k)}$ is the degraded[5] value computed in (9.30), both expressed in the coordinate system of lamina k.

[5] Homogenized.

The coefficient of thermal expansion of the cracking lamina k is calculated in a similar fashion, as follows

$$\alpha^{(k)} = \frac{1}{h_k} S^{(k)} \left(h \; Q \; \alpha - \sum_{m \neq k} h_m Q^{(m)} \alpha^{(m)} \right) \tag{9.33}$$

with $S = \left[Q^{(k)} \right]^{-1}$. The corresponding thermal damage is calculated as

$$D_j^{\alpha(k)} = 1 - {\alpha_j}^{(k)} / \widetilde{\alpha}_j^{(k)} \quad ; \quad j = 2, 6 \tag{9.34}$$

9.7 Fracture Energy

Under displacement control, the energy release rate (ERR) is defined as the partial derivative of the strain energy U with respect to the crack area A (see (10.1)). According to experimental observations on laminated, brittle matrix composites (e.g., using most toughened epoxy matrices), cracks develop suddenly over a finite length, and thus are not infinitesimal. Then, Griffith's energy principle is applied on its discrete (finite) form in order to describe the observed, discrete (finite) behavior of crack growth, as follows

$$\begin{aligned} G_I &= -\frac{\Delta U_I}{\Delta A} \\ G_{II} &= -\frac{\Delta U_{II}}{\Delta A} \end{aligned} \tag{9.35}$$

where $\Delta U_I, \Delta U_{II}$ are the change in laminate strain energy during mode I and mode II finite crack growth, respectively; and ΔA is the newly created (finite) crack area, which is one half of the new crack surface. Counting crack area as one-half of crack surface is consistent with the classical fracture mechanics convention for which fracture toughness G_c is twice of Griffith's surface energy γ_c.

To calculate the ERR, it is convenient to use the laminate stiffness Q in the c.s. of the cracked lamina, because in this way, the ERR can be decomposed into opening and shear modes. Since the laminate stiffness is available from the analysis as a function of crack densityλ, the ERR can be calculated, for a fixed strain level (load), and using [110] and [142, Section 3.2.10], into (9.35), we arrive at

$$G_I = -\frac{V}{2\Delta A} \left(\epsilon_2 - \alpha_2 \Delta T \right) \Delta Q_{2j} \left(\epsilon_j - \alpha_j \Delta T \right) \quad ; \quad \text{opening mode} \tag{9.36}$$

$$G_{II} = -\frac{V}{2\Delta A} \left(\epsilon_6 - \alpha_6 \Delta T \right) \Delta Q_{6j} \left(\epsilon_j - \alpha_j \Delta T \right) \quad ; \quad \text{shear mode} \tag{9.37}$$

where $V, \Delta A$ are the volume of the RVE and the increment of crack area, respectively; ΔQ_{ij} is the change in laminate stiffness corresponding to the change in crack area experienced; and all quantities are *laminate average* quantities expressed in the c.s of the cracked lamina in order to allow for ERR mode decomposition [110].

In the current implementation of the model, which is used in Example 9.1 (p. 341), $\Delta A = 1/h_k$ is the area of one new crack appearing halfway between two existing cracks. In this case the crack density doubles and $\Delta Q = Q(2\lambda) - Q(\lambda) < 0$. Alternative crack propagation strategies are considered in [126].

It can be seen that the proposed methodology provides the key ingredients for the computation of the ERR; namely the degraded stiffness and degraded CTE of the laminate, both as a function of crack density.

The damage activation function (9.1) can now be calculated for any value of λ and applied strain $\epsilon_x, \epsilon_y, \gamma_{xy}$. Note that the computation of the ERR components derives directly from the displacement solution (9.16) for a discrete crack (Figure 9.3). When this formulation is used along with the finite element method (FEM), it does not display mesh dependency on the solution and does not require the arbitrary specification of a characteristic length [137], in contrast to formulations based on smeared crack approximations [55]. The effect of residual thermal stresses is incorporated into the formulation. The code is available as a user material for *ANSYS* [137], a shell user element for *ANSYS* [138], and a user general section (UGENS) for *Abaqus* [140]; see Example 9.1, p. 341.

9.8 Solution Algorithm

The solution algorithm consists of (a) strain steps, (b) laminate-iterations, and (c) lamina-iterations. The state variables for the laminate are the array of crack densities for all laminas i and the membrane strain ϵ. At each load (strain) step, the strain on the laminate is increased and the laminas are checked for damage.

9.8.1 Lamina Iterations

When matrix cracking is detected in lamina k, a return mapping algorithm (RMA) (Section 8.4.1) is invoked to iterate and adjust the crack density λ_k in lamina k in such a way that g_k returns to zero while maintaining equilibrium between the external forces and the internal forces in the laminas. The iterative procedure works as follows. At a given strain level ϵ for the laminate and given λ_k for lamina k, calculate the value of the damage activation function g_k and the damage variables, which are both functions of λ_k. The RMA calculates the increment (decrement) of crack density as

$$\Delta\lambda_k = -g_k / \frac{\partial g_k}{\partial \lambda} \tag{9.38}$$

until $g_k = 0$ is satisfied within a given tolerance, for all $k = 1...N$, where N is the number of laminas in the laminate. The analysis starts with a negligible value of crack density present in all laminas ($\lambda = 0.02$ cracks/mm were used in the examples).

9.8.2 Laminate Iterations

To calculate the stiffness reduction of a cracked lamina (k-lamina), all of the other laminas (m-laminas) in the laminate are considered not damaging during the course

Table 9.1: Properties for Example 9.1.

Property	Value
E_1 [GPa]	44.7
E_2 [GPa]	12.7
G_{12} [GPa]	5.8
ν_{12}	0.297
ν_{23}	0.41
Ply thickness [mm]	0.144
G_{Ic} $[kJ/m^2]$	0.254
G_{IIc} $[kJ/m^2]$	-
CTE$_1$ [1E-6/C]	3.7
CTE$_2$ [1E-6/C]	30
ΔT[C]	0

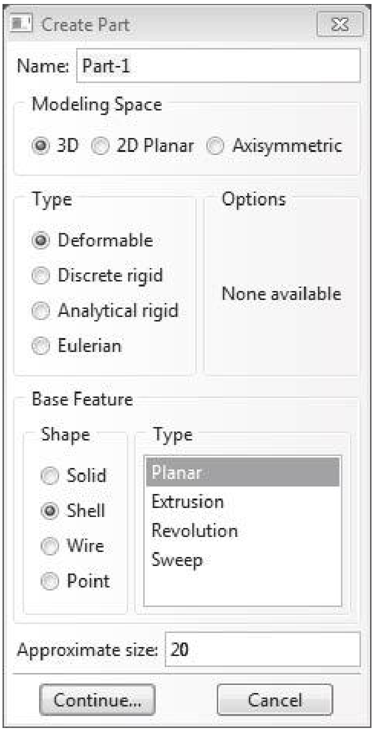
of lamina-iterations in lamina k, but with damaged properties calculated according to the current values of their damage variables $D_i^{(m)}$. Given a trial value of λ_k, the analytical solution provides $g_k, D_i^{(k)}$ for lamina k assuming all other laminas do not damage while performing lamina iterations in lamina k. Since the solution for lamina k depends on the stiffness of the remaining laminas, a converged iteration for lamina k does not guarantee convergence for the same lamina once the damage in the remaining laminas is updated. In other words, within a given strain step, the stiffness and damage of all the laminas are interrelated and they must all converge. This can be accomplished by laminate-iterations; that is, looping over all laminas repeatedly until all laminas converge to $g = 0$ for all k. Unlike classical RMA set up for plasticity, where the hardening parameter is monotonically increasing (Section 8.4.1), the crack density λ_k must be allowed to decrease if in the course of laminate iterations, other laminas sustain additional damage that makes the laminate more compliant and thus requires a reduction of λ_k.

Example 9.1 *Consider a* $[0/90_8/0/90_8/0]$ *laminate made of Glass/Epoxy with properties given in Table 9.1 subjected to a membrane strain* $\epsilon_x \neq 0, \epsilon_y = \gamma_{xy} = 0$. *Plot crack density in lamina* $k = 2$ *vs. strain* $0 < \epsilon_x < 2\%$.

Solution to Example 9.1 *Since* Abaqus *does not have built in capability for calculating crack density, we have to use a plugin. In this case, we use a programmable User General Section (UGENS, [143, Section 26.6.6]) that implements Discrete Damage Mechanics. The DDM plugin is available in [5, ugens-std.obj] and the theory behind it is explained in this chapter.*

This example illustrates how to create a model using Abaqus/*CAE; write an* **`.inp`** *file; modify the* **`.inp`** *file to add the data needed by the UGENS; run the Job with the plugin; and finally visualize the results. The procedure is illustrated using pseudo code (review Example 2.3, p. 54, if needed) and screen shots of the GUI.*

i. Starting Abaqus/*CAE from a command window (command shell)*
Since we need a number of files to be readily available, we start Abaqus/*CAE*



Name: Part-1

Modeling Space

● 3D ○ 2D Planar ○ Axisymmetric

Type

● Deformable
○ Discrete rigid
○ Analytical rigid
○ Eulerian

Options

None available

Base Feature

Shape

○ Solid
● Shell
○ Wire
○ Point

Type

Planar
Extrusion
Revolution
Sweep

Approximate size: 20

Fig. 9.4: Create the part.

in the folder (directory) of our choice. Right click on[6] *the folder* ***`c:\SIMULIA\User\Chapter_9\Ex_9.1`*** *and choose* ***`Open command window here`****. In the command window, type:* ***`abaqus cae`****. This will open* Abaqus/*CAE with the* Work Directory *set to the current folder.*

ii. *Creating the part*
Since the strain field inside the shell is uniform (in x and y), we use only one S4R element to model a unit cell with dimensions $a = 10 \times b = 10$ mm, where a, b are the dimensions of the model along the x, y, directions respectively. The pseudo code for this example begins as follows:

```
Module: Part
Menu:   Part, Create, Name; [Part-1], 3D, Deformable, Shell, Planar,
             Approx. size: [20]
        Add, Lines, Rectangle, [0,0], [10,10], X, Done
```

The first line in the pseudo code above is equivalent to Figure 9.4, and so on.

iii. *Entering the section properties*
With UGENS, the material properties and section parameters must be entered

[6] Do not place the pointer (cursor) *in* the folder, but *on* the yellow icon for the folder.

as section properties*, but as of release 6-10,* Abaqus/*CAE does not have capability to enter the data for UGENS. Instead, one has to edit the* `Job-1.inp` *file, similarly to Example 4.3, p. 188. Before doing that, we use CAE to enter the parameters as if they were to be used for a user material. The DDM plugin employs* $4 + 9N + 3N$ *parameters, as follows:*

- *The (total) laminate thickness* t.
- $3 + 9 * N$ *material properties, where* N *is the number of laminas in the symmetric part of the laminate, i.e., in one half of the LSS. The properties are ordered as follows, starting with the first lamina,* $k = 1$ *(bottom surface), and continuing until the lamina* N *(middle surface):*

  G_{Ic} *Critical value of ERR in mode I.*

  G_{IIc} *Critical value of ERR in mode II. Use* $G_{IIc} > 4\, G_{Ic}$ *if data is not available.*

  ΔT *Change in temperature from the temperature at which* G_{Ic}, G_{IIc}*, were measured to the operating temperature.*

  E_1 *Longitudinal modulus.*

  E_2 *Transverse modulus.*

  G_{12} *In-plane shear modulus.*

  ν_{12} *In-plane Poisson's ratio.*

  ν_{23} *Intralaminar Poisson's ratio. Note;* $G_{23} = E_2/2/(1 + \nu_{23})$.

  α_1 *Longitudinal coefficient of thermal expansion.*

  α_2 *Transverse coefficient of thermal expansion.*

  θ_k *Lamina orientation with respect to the laminate c.s.*

  t_k *Lamina thickness.*
- $3 * N$ *state variables, starting with the first lamina,* $k = 1$ *(bottom surface), and continuing until the lamina* N *(middle surface):*

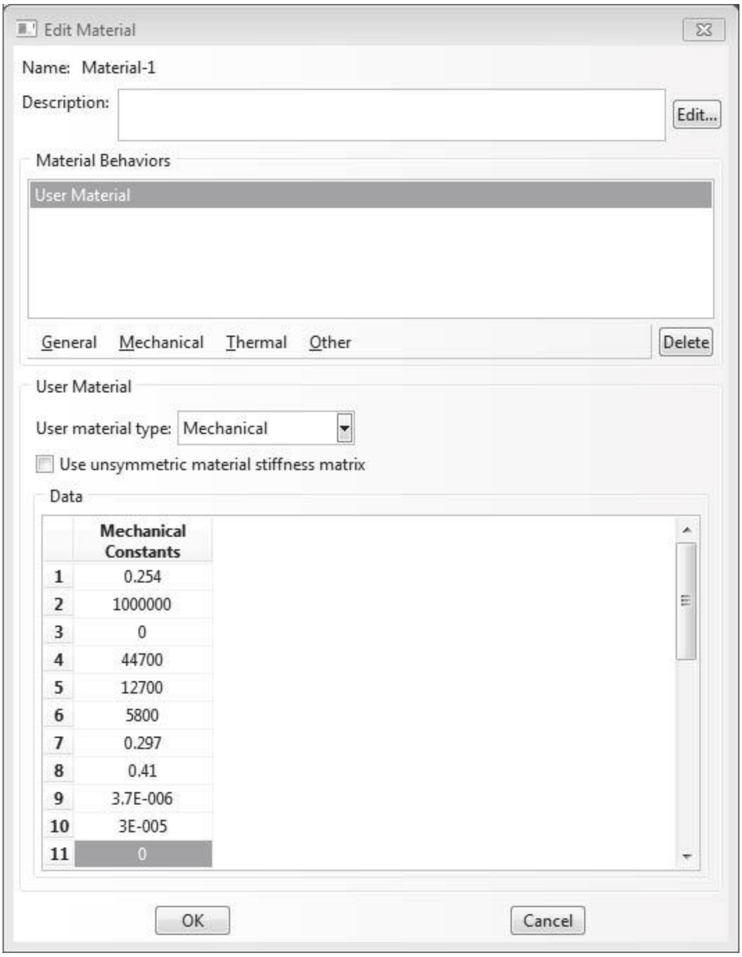
  λ_k *Crack density in lamina k.*

  D_2 *Transverse damage, lamina k.*

  D_6 *Shear damage, lamina k.*

Enter $3+9N+5N$ *values (all except the laminate thickness) into the Material as shown in Figure 9.5. The values are given in Step viii.*

Then, create a section and specify the shell thickness (see Figure 9.6) and three values for the intralaminar shear coefficients (Figure 9.7). Since in this example there is not bending and no transverse shear, the value of the intralaminar shear coefficients is irrelevant; one just has to enter something. Otherwise, the intralaminar shear coefficients can be calculated as in (3.9) *(see also Example 3.1, p. 108). Lastly, assign the section to the part (*`Menu: Assign`*) and Instance the Part (*`Module: Assembly`*). The pseudo code describes the process to be followed using the GUI in* Abaqus/*CAE in detail:*



Name: Material-1

Description:

Material Behaviors

User Material

User Material

User material type: Mechanical

☐ Use unsymmetric material stiffness matrix

Data

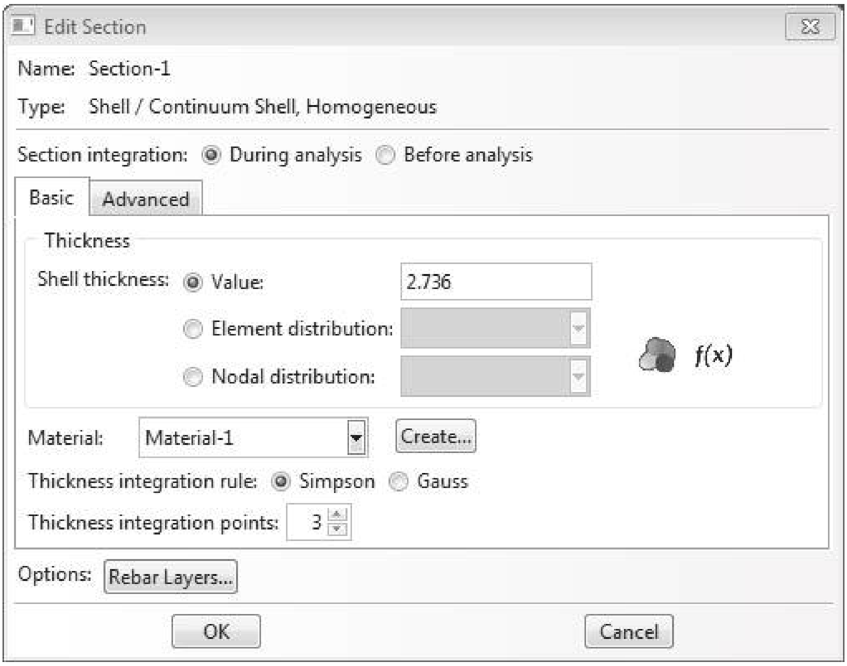
	Mechanical Constants
1	0.254
2	1000000
3	0
4	44700
5	12700
6	5800
7	0.297
8	0.41
9	3.7E-006
10	3E-005
11	0

Fig. 9.5: Create the material.

```
Module: Property
Menu:   Material, Create, Name: [Material-1],
            Material Behavior: User Material,
            Mechanical Constants: # 3+9*N+5*N values
        Section, Create, Name: [Section-1], Shell, Homogeneous, Cont
            Tab: Basic, Shell thickness Value; [2.736],
                Material: Material-1
            Tab: Advanced: Transverse Shear Stiffness: [1,0,1]
        Assign, Section, # select the part, Done, OK
Module: Assembly
Menu:   Instance, Create, Name: [Part-1], Type: Independent, OK
```

iv. *Creating a step*

Next, a step is needed, among other things, to set the ***Field Output Requests*** *(See Figure 9.8). Since the analysis has material nonlinearity (damage), one has to be careful with the solution strategy (see Figure 9.9). This is accom-*



Edit Section

Name: Section-1

Type: Shell / Continuum Shell, Homogeneous

Section integration: ● During analysis ○ Before analysis

Basic | Advanced

Thickness

Shell thickness: ● Value: 2.736

○ Element distribution:

○ Nodal distribution:

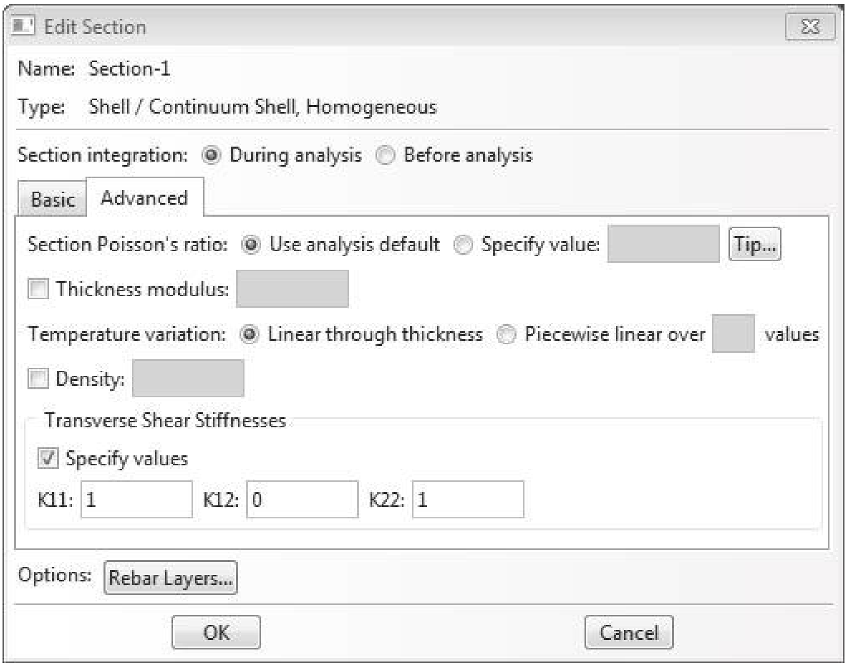
Material: Material-1

Thickness integration rule: ● Simpson ○ Gauss

Thickness integration points: 3

Options: Rebar Layers...

Fig. 9.6: Define the section.



Edit Section

Name: Section-1

Type: Shell / Continuum Shell, Homogeneous

Section integration: ● During analysis ○ Before analysis

Basic | Advanced

Section Poisson's ratio: ● Use analysis default ○ Specify value:

□ Thickness modulus:

Temperature variation: ● Linear through thickness ○ Piecewise linear over values

□ Density:

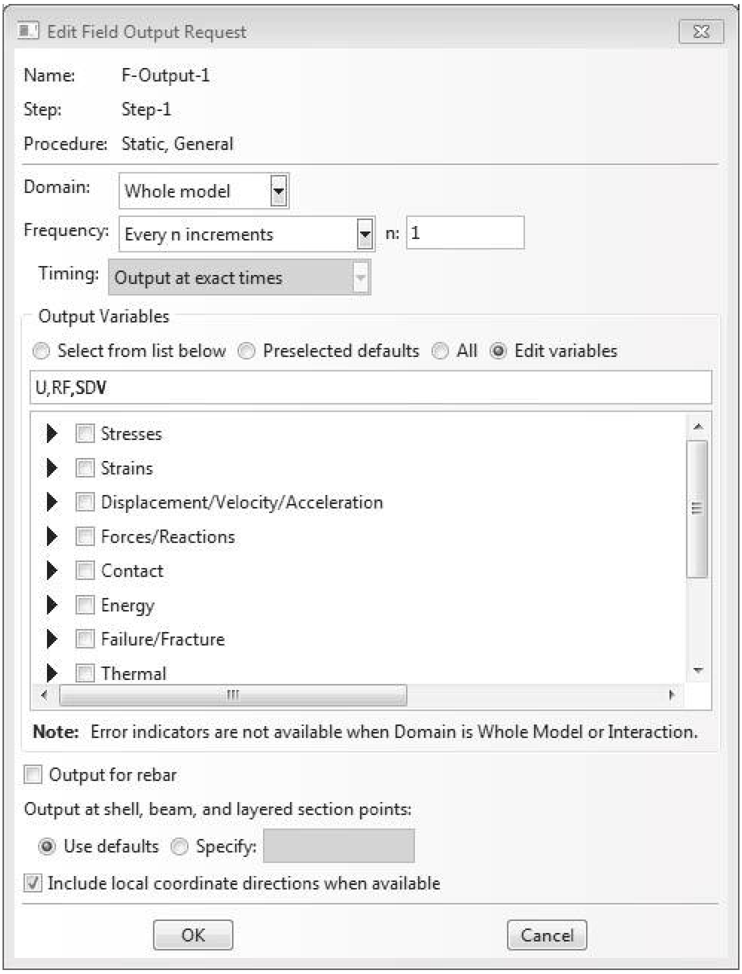
Transverse Shear Stiffnesses

☑ Specify values

K11: 1 K12: 0 K22: 1

Options: Rebar Layers...

Fig. 9.7: Define the section, continued.



Edit Field Output Request

Name: F-Output-1

Step: Step-1

Procedure: Static, General

Domain: Whole model

Frequency: Every n increments n: 1

Timing: Output at exact times

Output Variables

○ Select from list below ○ Preselected defaults ○ All ● Edit variables

U,RF,SDV

- □ Stresses
- □ Strains
- □ Displacement/Velocity/Acceleration
- □ Forces/Reactions
- □ Contact
- □ Energy
- □ Failure/Fracture
- □ Thermal

Note: Error indicators are not available when Domain is Whole Model or Interaction.

□ Output for rebar

Output at shell, beam, and layered section points:

● Use defaults ○ Specify:

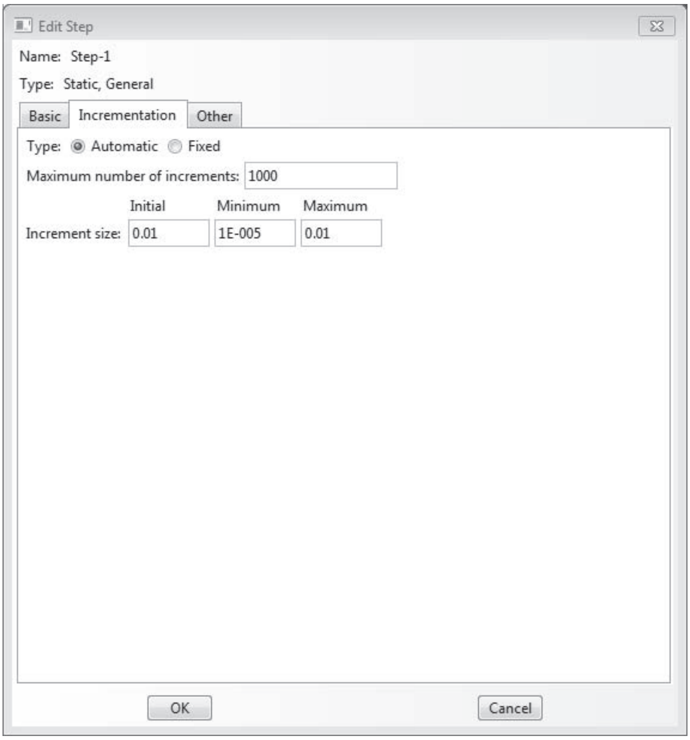
☑ Include local coordinate directions when available

Fig. 9.8: Field Output Request.

plished by following the pseudo code:

```
Module: Step
Menu: Step, Create, Name: [Step-1], After: Initial,
        Type: General, Static, General, Cont
        Tab: Basic, Time period: 1
        Tab: Advanced, max. increments: [1000],
        Incr.size: Initial: [0.01], Minimum; [1E-005], Maximum: [0.01]
```

v. Specifying boundary conditions
Next comes the boundary conditions, with symmetry along $x = 0$ and $y = 0$. The load is applied as a displacement, with a maximum value calculated as the product of the x-dimension of the model times the maximum strain expected, i.e., $U_1 = a \max \epsilon_x = 0.2$ (see Figure 9.10). The details are given by the pseudo code:



Name: Step-1

Type: Static, General

Type: ● Automatic ○ Fixed

Maximum number of increments: 1000

	Initial	Minimum	Maximum
Increment size:	0.01	1E-005	0.01

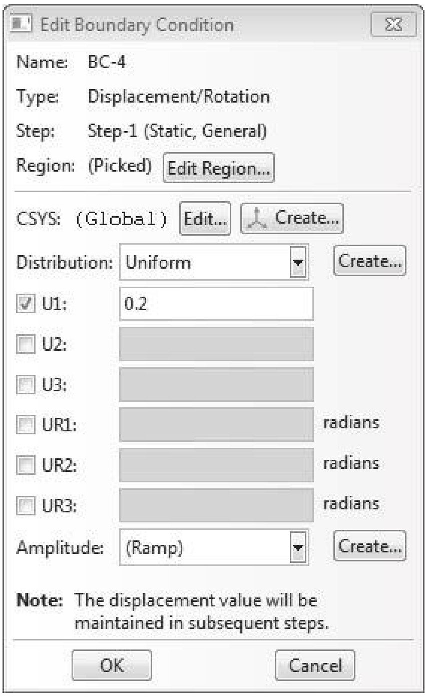
Fig. 9.9: Define the step.

```
Module: Load
Menu:   BC, Manager
        Create, name: [BC-1], Step: Initial, Type: Symmetry, Cont,
            # select vertical edge @ x=0, Done, XSYMM, OK
        Create, name: [BC-2], Step: Initial, Type: Symmetry, Cont,
            # select horizontal edge @ y=0, Done, YSYMM, OK
        Create, name: [BC-3], Step: Initial, Type: Symmetry, Cont,
            # select vertex at the origin, Done, ENCASTRE, OK
        Create, name: [BC-4], Step: Step-1, Type: Displacement, Cont,
            # select vertical edge @ x=10, Done, U1=[0.2], OK
```

vi. Meshing
Then, mesh with elements S4R. Since the membrane strain will be uniform, only one element is needed. This is accomplished by selecting a seed size equal to the model dimensions $a = b$. For example, follow this procedure:

```
Module: Mesh
Menu:   Seed: Instance, Approx. size: [10], OK
        Mesh, Element Type, Standard, Linear, Shell # S4R
        Mesh, Instance, Yes
```

vii. Creating the first Job
Next, create a job just to print the `Job-1.inp` file (see Figure 9.11).



Name: BC-4
Type: Displacement/Rotation
Step: Step-1 (Static, General)
Region: (Picked) Edit Region...
CSYS: (Global) Edit... Create...
Distribution: Uniform Create...
☑ U1: 0.2
□ U2:
□ U3:
□ UR1: radians
□ UR2: radians
□ UR3: radians
Amplitude: (Ramp) Create...

Note: The displacement value will be maintained in subsequent steps.

Fig. 9.10: Boundary Conditions.

```
Module: Job
Menu:   Job, Manager
        Create, Name: [Job-1], Source: Model, Cont, OK
        Data Check
        Write Input
```

viii. *Modifying the* `.inp` *file*

Next copy `Job-1.inp` *as* `Job-ddm.inp` *and modify it as follows. Both files are available in [5].*

Replace the following line:

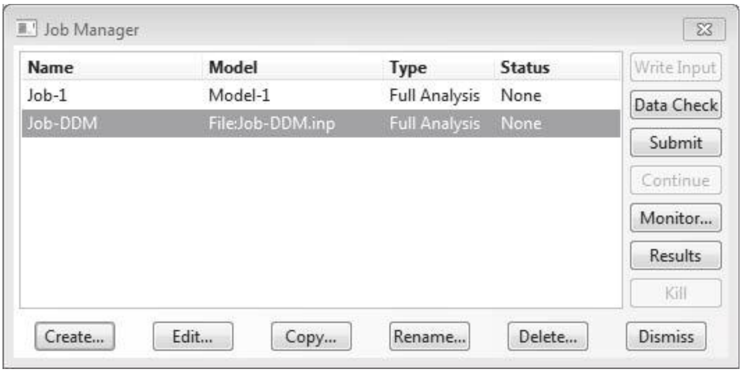
```
*Shell Section, elset=_PickedSet2, material=Material-1
```

by the follwing single line (shown as two lines below due to space constraints, but it must be a single line in the `.inp` *file) :*

```
*Shell General Section,
elset=_PickedSet2, USER, PROPERTIES=30, VARIABLES=15
```

Take the material properties:

```
  0.254,   1e+06,      0., 44700., 12700.,  5800.,  0.297,  0.41
3.7e-06,   3e-05,      0.,  0.144, 44700., 12700.,  5800., 0.297
   0.41, 3.7e-06,   3e-05,    90.,  1.152, 44700., 12700., 5800.
  0.297,    0.41, 3.7e-06,  3e-05,     0.,   0.072
```



Name	Model	Type	Status
Job-1	Model-1	Full Analysis	None
Job-DDM	File:Job-DDM.inp	Full Analysis	None

Fig. 9.11: Create the first Job (Job-1) using the Job Manager.

and move them immediately above the following line:

```
*Transverse Shear
```

Delete the following lines:

```
** MATERIALS
**
*Material, name=Material-1
*User Material, constants=30
**
```

Add the following (call to an internal routine that sets the initial crack density to $\lambda = 0.02$*):*

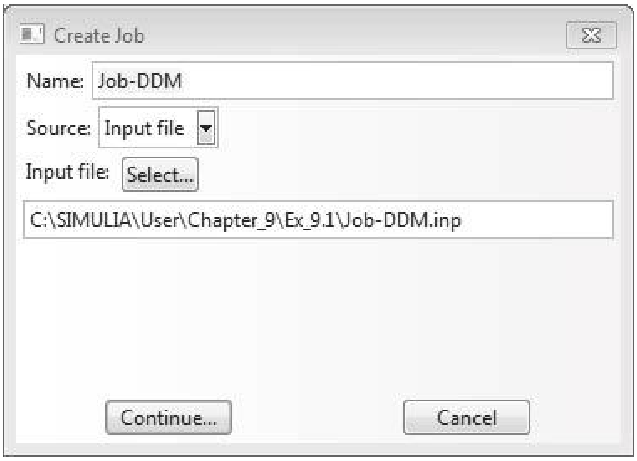
```
*Initial Conditions, type=SOLUTION, USER
```

just before the following line:

```
** STEP: Step-1
```

ix. *Creating a second Job*
Next, create another job (`Job-DDM`*) to run the model with the plugin (see Figures 9.12,9.13):*

```
Module: Job
Menu:   Job, Manager
    Create, Name: [Job-DDM], Source: Input file,
    Select: [C:\SIMULIA\User\Chapter_9\Ex_9.1\Job-DDM.inp], Cont,
    Tab: General,
    User subroutine: [C:\SIMULIA\User\Chapter_9\Ex_9.1\ugens-std.obj],
    OK
    Submit # wait for the run to complete
    Results
```



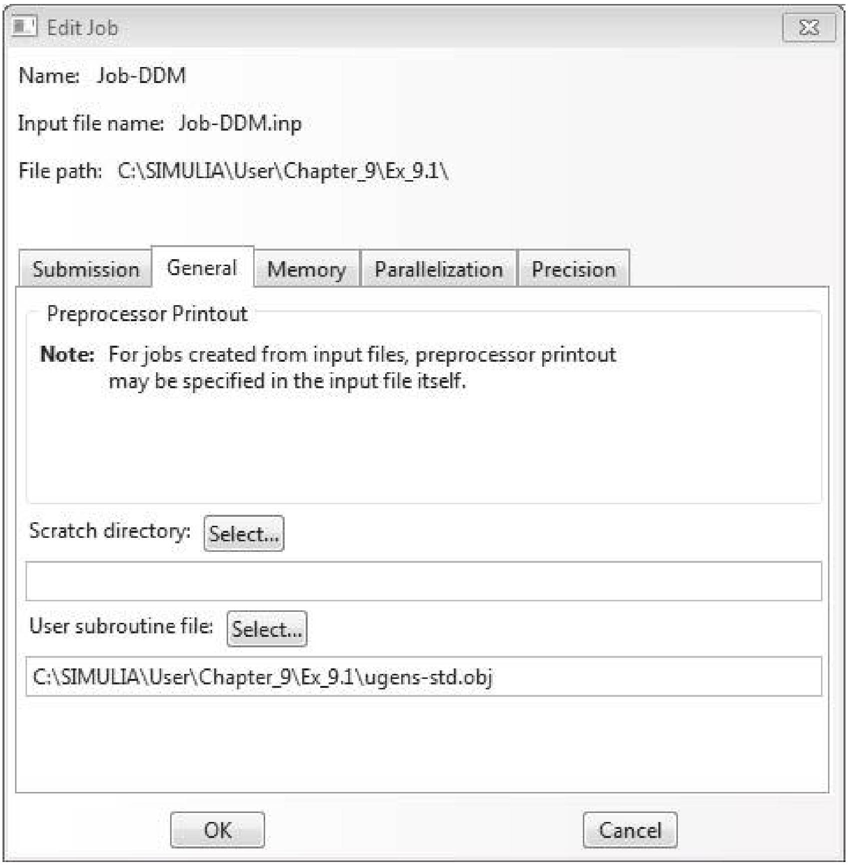
Name: Job-DDM

Source: Input file

Input file: Select...

C:\SIMULIA\User\Chapter_9\Ex_9.1\Job-DDM.inp

Fig. 9.12: Add a second job with source type: Input File.



Name: Job-DDM

Input file name: Job-DDM.inp

File path: C:\SIMULIA\User\Chapter_9\Ex_9.1\

Note: For jobs created from input files, preprocessor printout may be specified in the input file itself.

Scratch directory: Select...

User subroutine file: Select...

C:\SIMULIA\User\Chapter_9\Ex_9.1\ugens-std.obj

Fig. 9.13: Edit the second job to specify the user subroutine file.

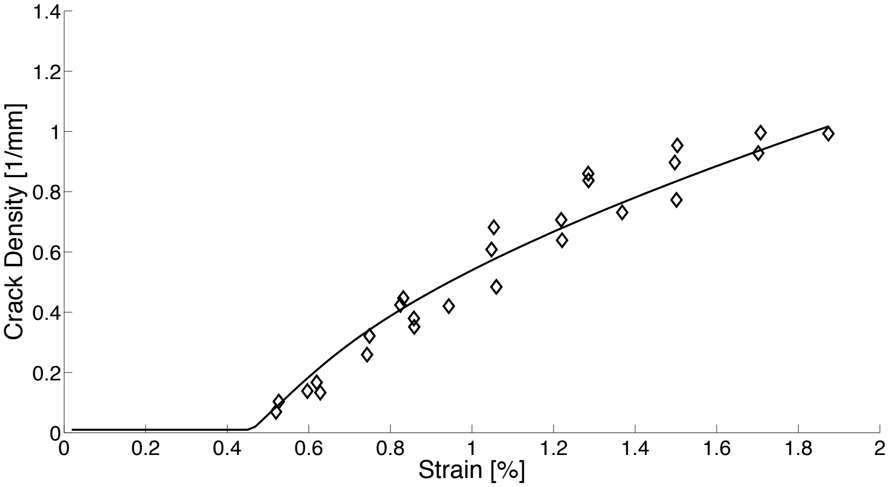


Fig. 9.14: Crack density vs. strain predicted by DDM and compared to experimental data from [144].

x. *Visualization*
To visualize, recall the meaning of the state variables (see Step iii). For this laminate, lamina #2 is cracking. The crack density λ_2 is stored in SDV=4. The initial value is set to $\lambda = 0.02$. Select step #0 to see the initial value. At step #24 ($\epsilon_x \approx 0.48\%$) the first crack appears.

```
Module: Visualization
Menu:   Plot, Contours, On deformed shape
        Result, Step/Frame, # select step 24
```

Alternatively, the whole procedure can be automated by using a couple of Python scripts, `Ex_9.1.py` and `UgenKeyword.py`, available in [5]. To execute the script, open a command window as explained in step i, and type: `abaqus cae nogui=ex_9.1`. Then, open the `Job.odb` to visualize the results, and/or use the results saved on the `Job.csv` file. A plot of crack density vs. strain is shown in Figure 9.14.

Suggested Problems

Problem 9.1 *Calculate the critical laminate strain ϵ_x^c for which the first crack appears in laminates 1 to 3 in Table 9.2, all made of Glass/Epoxy with properties given in Table 9.1 subjected to a membrane strain $\epsilon_x \neq 0, \epsilon_y = \gamma_{xy} = 0$. For each laminate, in what lamina does the first crack appear?*

Problem 9.2 *Using the results from Exercise 9.1, calculate the in situ transverse strength F_{2t}^{is} of the laminas [1, Section 7.2.1].*

Table 9.2: Laminates for Exercise 9.1.

Laminate Number	LSS
1	$[0/90_8/0/90_8/0]$
2	$[0/70_4/-70_4/0/-70_4/70_4/0]$
3	$[0/55_4/-55_4/0/-55_4/55_4/0]$
4	$[0_2/90_8/0_2]$
5	$[15/-15/90_8/-15/15]$
6	$[30/-30/90_8/-30/30]$
7	$[40/-40/90_8/-40/40]$

Problem 9.3 *Calculate the critical laminate strain ϵ_x^c for which the first crack appears in laminates 4 to 7 in Table 9.2, all made of Glass/Epoxy with properties given in Table 9.1 subjected to a membrane strain $\epsilon_x \neq 0, \epsilon_y = \gamma_{xy} = 0$. For each laminate, in what lamina does the first crack appear?*

Problem 9.4 *Using the results from Exercise 9.3, calculate the in situ transverse strength F_{2t}^{is} of the laminas [1, Section 7.2.1].*

Chapter 10

Delaminations

Delamination is a frequent mode of failure affecting the structural performance of composite laminates. The interface between layers offers a low-resistance path for crack growth because the bonding between two adjacent layers depends only on matrix properties. Delamination may originate from manufacturing imperfections, cracks produced by fatigue or low velocity impact, stress concentration near geometrical/material discontinuity such as joints and free edges, or due to high interlaminar stresses.

In laminates loaded in compression, the delaminated layers may buckle, and cracks propagate due to interaction between delamination growth and buckling. The presence of delaminations may reduce drastically the buckling load and the compressive strength of the composite laminates [145] (Figure 10.1). Delaminations may also be driven by buckling in laminates under transverse loading [146]. The analysis of delamination buckling requires the combination of geometrically nonlinear structural analysis with fracture mechanics.

According to its shape, delaminations are classified into through-the-width or strip [146–153], circular [153–159], elliptic [160], rectangular [161] or arbitrary [162, 163]. Depending on its location through the laminate thickness, delaminations are classified into thin film, symmetric split [145, 148, 149], and general [150, 153, 156, 157, 159]. In addition, analysis of combined buckling and growth for composite laminates containing multiple delaminations under in-plane compressive loading has been carried out [164, 165]. Experimental results on delamination buckling are presented in [166, 167].

Other delamination configurations that have been investigated in the literature are the beam-type delamination specimens subjected to bending, axial, and shear loading [166–172] which form the basis for experimental methods used to measure interlaminar fracture strength under pure mode I, mode II, and mixed mode conditions in composites, adhesive joints, and other laminated materials (Figure 10.2).

In plates with piezoelectric sensors or actuators, an imperfect bonding between the piezoelectric lamina and the base plate may grow under mechanical and/or electrical loading. As a consequence, the adaptive properties of the smart system can be significantly reduced since debonding results in significant changes to the static or dynamic response [173, 174]. Finally, delamination growth may be caused

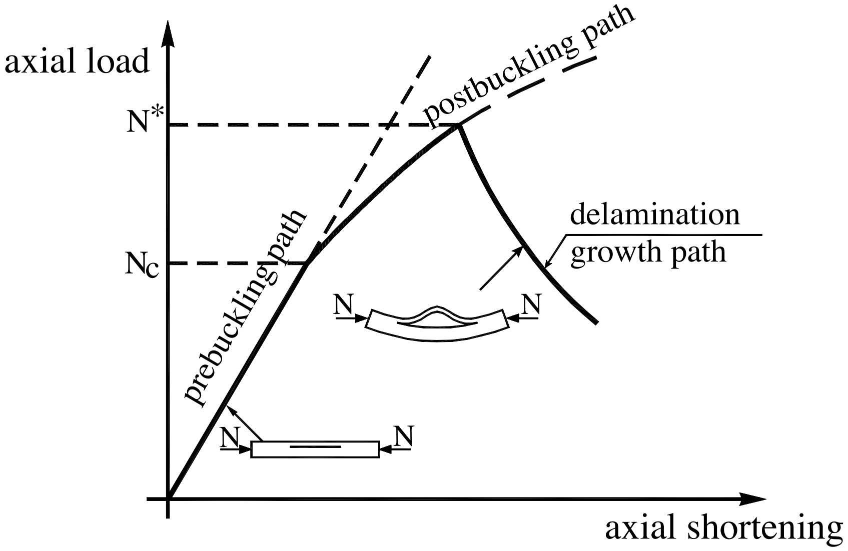


Fig. 10.1: Delamination buckling in a compressed laminate.

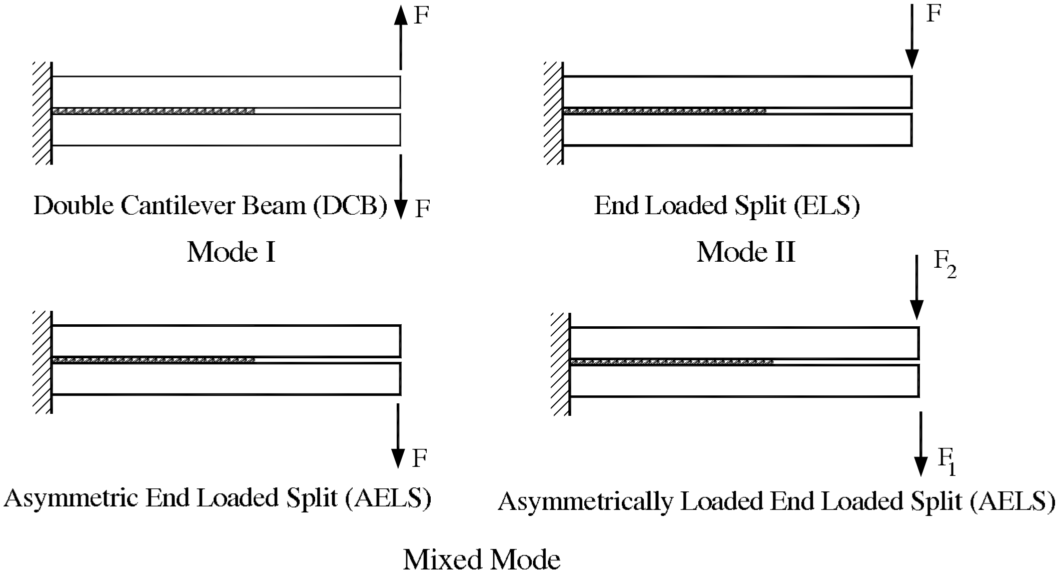

F
F
Double Cantilever Beam (DCB)
Mode I
F
End Loaded Split (ELS)
Mode II
F
Asymmetric End Loaded Split (AELS)
F_2
F_1
Asymmetrically Loaded End Loaded Split (AELS)
Mixed Mode

Fig. 10.2: Beam-type delamination specimens.

by dynamic effects, such as vibration and impact. For instance, the dynamics effects resulting from the inertia of the laminate on the growth process resulting from the buckling of the delamination have been investigated for a circular delamination and time-dependent loadings [175].

Delaminations can be analyzed by using *cohesive damage models* (Section 10.1) and *fracture mechanics* (Section 10.2). A cohesive damage model implements interfacial constitutive laws defined in terms of damage variables and a damage evolution law. Cohesive damage elements are usually inserted between solid elements [176–179] or beam/shell elements [178].

In the fracture mechanics approach, the propagation of an existing delamination is analyzed by comparing the amount of energy release rate (ERR) with the fracture toughness of the interface. When mixed mode conditions are involved, the decomposition of the total ERR into mode I, mode II, and mode III components becomes necessary due to the mixed-mode dependency of interface toughness [170, 180]. A number of fracture mechanics-based models have been proposed in the literature to study delamination, including three-dimensional models [181–183] and simplified beam-like models [145, 147, 172, 184, 185].

Fracture mechanics allows us to predict the growth of a preexisting crack or defect. In a homogeneous and isotropic body subjected to a generic loading condition, a crack tends to grow by kinking in a direction such that a pure mode I condition at its tip is maintained. On the contrary, delaminations in laminated composites are constrained to propagate in its own plane because the toughness of the interface is relatively low in comparison to that of the adjoining material. Since a delamination crack propagates with its advancing tip in mixed mode condition, the analysis requires a fracture criterion including all three mode components (Section 10.1.2).

The elastic strain energy per unit volume (density, in J/m^3) is defined as $U_0 = 1/2\sigma_{ij}\epsilon_{ij}$. The strain energy (in J) is defined as the volume integral $U = \int_V U_0 dV$. The energy required to form, or to propagate, a crack is equal to the elastic energy released by the solid during crack formation. The energy released is the difference between the elastic strain energy available before and after the crack is formed, i.e., $-\Delta U = U_{after} - U_{before}$. The rate of energy released per unit of crack area A is given, in J/m^2, by

$$G = -\frac{\Delta U}{\Delta A} \tag{10.1}$$

where A is one half the surface area created. The theory of crack growth may be developed by using one of two approaches due to Griffith and Irwin, respectively. The Griffith energy approach uses the concept of Energy Release Rate G as the (computable) energy available for fracture on one hand, and the material property G_c, which is the energy necessary for fracture, on the other hand. A crack grows when

$$G \geq G_c \tag{10.2}$$

where for completeness note that $G_c = 2\gamma_c$, where γ_c is the critical fracture energy per unit surface crack area, and A is one half of the crack area formed, i.e., the area

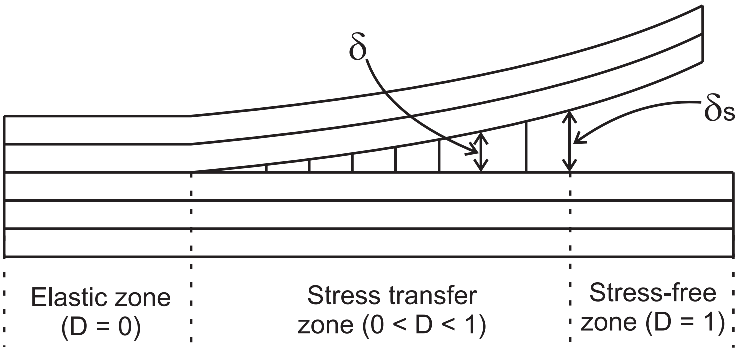


Fig. 10.3: Cohesive zone method to simulate crack propagation.

of only one of the faces of the crack.

The Irwin (local) approach is based on the concept of stress intensity factor, which represents the energy stress field in the neighborhood of the crack tip. These two approaches are equivalent and, therefore, the energy criterion may be rewritten in terms of stress intensity factors. Further, a number of path independent integrals have been proposed to calculate the ERR, such as the J-integral [186].

The elastic strain energy released ΔU during crack propagation, and therefore used to create the new surface area, can be calculated as the work required to close the crack, i.e.,

$$\Delta U = W_{closure} \tag{10.3}$$

The crack closure method of computation provides the base for the Virtual Crack Closure Technique (VCCT) described in Section 10.2.

10.1 Cohesive Zone Method

The cohesive zone method (CZM) is based on the assumption that the stress transfer capacity between the two separating faces of a delamination is not lost completely at damage initiation, but rather is a progressive event governed by progressive stiffness reduction of the interface between the two separating faces (Fig. 10.3).

The interface between the two possible separating faces of the laminated material is modeled with *cohesive elements* [187]. There are two types of CZM elements in *Abaqus*:

- CZM elements with finite thickness, which are based on a regular, continuum ($\sigma - \epsilon$) constitutive behavior. This type of elements is used to model delamination behavior of adhesive bonds where the thickness of the interface is considerable.
- CZM elements with zero thickness, which are based on a traction–separation ($\sigma - \delta$) constitutive behavior. This type of elements is used to model delamination behavior of adhesive bonds where the thickness of the interface is negligible, as it is the case of laminated composites.

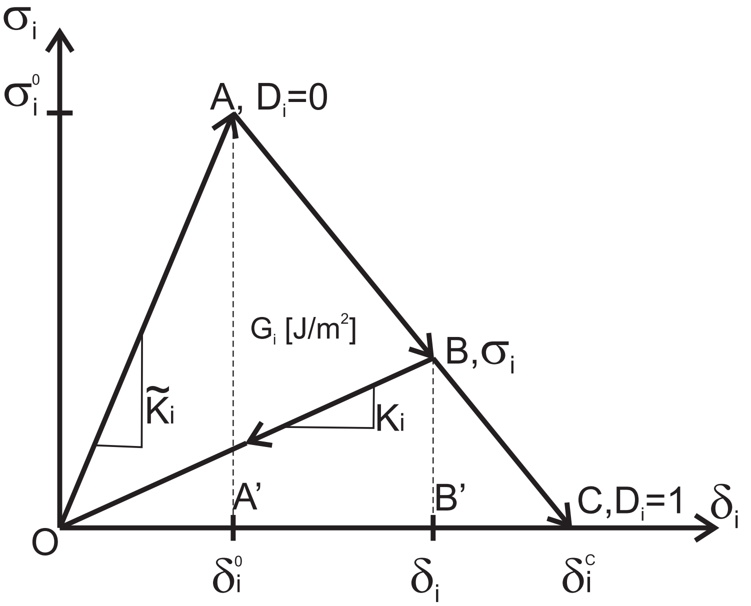


Fig. 10.4: Stress transfer model for cohesive zone model.

The cohesive behavior is described in terms of a traction-separation equation (Fig. 10.4). As the name implies, this approach replaces the engineering stress-strain ($\sigma - \epsilon$) equation with a traction-separation ($\sigma - \delta$) equation. The thickness of the element is set to zero by defining coincident opposite nodes of the cohesive element. However, even if the opposite nodes are initially coincident, they are still separate entities, and they separate during the deformation of the laminated composite. The adjacent faces of the laminas can be regarded as being connected to each other through the stiffness of the cohesive element. During the deformation, the resulting separation between adjacent laminas is proportional to the stiffness of the cohesive element.

The CZM element with finite thickness COH2D4 (Example 10.1, p. 364) and the CZM element with zero thickness, which is implemented through *interaction cohesive behavior* (Example 10.2), are based on a traction-separation constitutive equation. The element stiffness matrix requires the stiffness $\widetilde{K}$ of the interface material (called *penalty* stiffness in the *Abaqus* documentation), but the element stiffness matrix is not formulated as usual by integration over the volume of the element because the initial volume of the element is zero. The CZM element can be visualized as a spring between the initially coincident nodes of the element. However, the stiffness of the CZM element is part of the structural stiffness, and the CZM element will undergo deformation during the loading of the laminate. The initially coincident nodes will open (mode I: opening) or slide (model II: shear and III: tearing) relative to each other (Figure 10.5). The nodal separations of the CZM element are always known by solving the discretized structure.

Since the initial thickness of the CZM element is zero, the deformation state of the CZM element can not be described by the classical definition of strain. Instead,

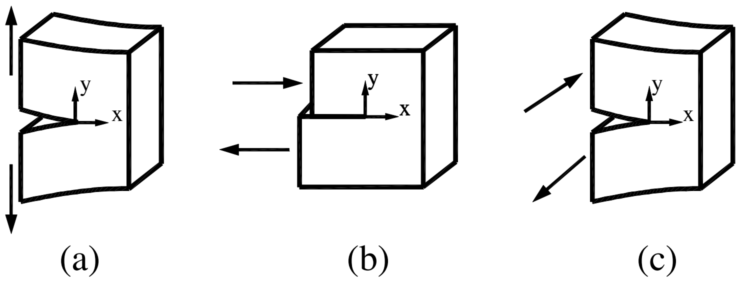


Fig. 10.5: Crack propagation modes: a) I-opening, b) II-shear, c) III: tearing.

the measure of the deformation becomes the separation δ between the faces connected through the CZM element, and this makes possible the use of the $(\sigma - \delta)$ traction-separation equation instead of the classical engineering $(\sigma - \epsilon)$ equation.[1]

10.1.1 Single Mode Cohesive Model

The CZM is formulated assuming the three crack propagation modes are uncoupled, even if multiple modes are active simultaneously, as described in Section 10.1.2. In this section we consider the case of a single mode deformation at an interface of the laminated material, either mode I, II, or III (Figure 10.5). The formulation is similar for any of the three modes. The surface tractions at the interface are σ_i with $i = I, II, III$ denoting the three modes of crack propagation. The corresponding separations between the opposite faces of the CZM element are denoted as δ_i, each related to the surface traction σ_i through the interface stiffness K_i, also called *penalty stiffness*. Therefore, material stiffness values are necessary for each of the loading modes, i.e., K_I, K_{II}, K_{III}. A discussion on how to choose numerical values for K_i is provided in [188].

The behavior of the material at the interface is assumed to be linear-elastic up to the onset of damage (OA in Fig. 10.4) and damaging-elastic after that (OB in Fig. 10.4). Consequently, the stress-separation is described by

$$\sigma_i = K_i \; \delta_i \tag{10.4}$$

and

$$K_i = (1 - D_i) \; \widetilde{K_i} \tag{10.5}$$

where D_i are the damage variables, and $\widetilde{K_i}$ in $[N/mm^3]$ are the stiffness values of the *undamaged* material at the interface, relating the stresses σ_i to the relative separations δ_i between the opposite faces of the CZM element, with $i = I, II, III$. The stiffness values $\widetilde{K_i}$ are additional material properties needed for the CZM model, which are different from the Young's modulus E for normal deformation, or G for shear deformation.

[1] *Abaqus* documentation uses the word *strain* as synonymous to *separation* δ when describing the deformation of the CZM element based on traction-separation equation.

For each mode, there is a critical level of stress σ_i^0 and separation δ_i^0 when the damage at the interface starts (point A in Fig. 10.4), called *damage onset.* At damage onset, the two laminas do not separate completely into a physical crack but rather the interface material starts losing its stiffness. Here, σ_i^0 represents the strength of the interface with one value of strength for each crack propagation mode: $\sigma_I^0, \sigma_{II}^0, \sigma_{III}^0$. In CZM, these modes are called *damage modes* because the CZM does not evaluate fracture mechanics. Instead, CZM replaces the fracture mechanics problem by a continuum damage mechanics problem (Chapter 8). The interface strengths are therefore additional material parameters required by the CZM element.

Therefore, the *damage initiation criteria* are

$$\sigma_i = \sigma_i^0 \tag{10.6}$$

and the separation at damage onset are calculated as

$$\delta_i^0 = \sigma_i^0 / \widetilde{K_i} \tag{10.7}$$

After damage onset, the interface material starts losing its stiffness (OB in Fig. 10.4), according to (10.5). Note that there are as many damage variables as damage modes: D_I, D_{II}, D_{III}, which are state variables (8.61) to be determined during the analysis. Their physical interpretation is given by (10.5) as measures of stiffness degradation (see also Section 8.2). The damage variables satisfy the following conditions:

- $D = 0$ up to damage onset (OA in Fig. 10.4) while the interface material is undamaged, thus retaining its initial stiffness.
- $0 < D < 1$ during degradation of the interface material (AC in Fig. 10.4), when the material is gradually losing its stiffness.
- $D = 1$ at fracture (point C in Fig. 10.4), when there is no remaining stiffness for the interface material, which means no stress transfer capacity is provided by the interface. This corresponds to the fracture of the cohesive connection between the two faces of the CZM element.

With reference to Figure 10.4 we have:

$$D_i = \begin{cases} 0, & \delta_i \leq \delta_i^0 \\ 1, & \delta_i = \delta_i^c \end{cases} \tag{10.8}$$

In stress-strain space, the cohesive behavior depicted in Fig. 10.4 displays *stress softening.* That is, after damage onset the stresses σ_i in the damaged interface have lower values than the peak values σ_i^0.

The CZM element uses the classical assumption of *elastic damage* typical of continuum damage mechanics (CDM, Chapter 8). Therefore, unloading from any point such as B on the line AF in Fig. 10.4 will return to the origin without permanent deformation upon unloading.

Eventually, there will be total fracture of the cohesive bond (point C in Fig. 10.4) when the stiffness of the interface reduces to zero. Due to the fact that total loss of stiffness, and thus total loss of cohesive stress transfer, does not take place until point C, the CZM is correlated with the Griffith crack propagation criterion (10.2) at point C. The correlation is made by considering that the area under the $(\sigma - \delta)$ curve in Fig. 10.4 is equal to the critical ERR G_{ic} in Griffith's principle (10.2). In this way, the separation at fracture δ_i^c can be calculated as

$$\delta_i^c = \frac{2G_{ic}}{\sigma_i^0} \tag{10.9}$$

Since there is a critical ERR value for each damage mode, G_{ic}, with $i = I, II, III$, there will be three separations at fracture–one per mode. The three ERR values are material properties required by the CZM, in addition to the three values of strength σ_i^0 and the three values of interface stiffness $\widetilde{K}_i$. From the point of view of the amount of experimental data needed to perform an analysis, this is a disadvantage of CZM because it requires nine experimental values. On the other hand, discrete damage mechanics requires only the three values of ERR to predict both the onset and the evolution of damage (Chapter 9).

Substituting (10.4) into (10.5) and rearranging, we get

$$D_i = 1 - \frac{\sigma_i \delta_i^0}{\sigma_i^0 \delta_i} \tag{10.10}$$

By similarity of triangles BB'F and AA'F, we have

$$\frac{\sigma_i}{\sigma_i^0} = \frac{\delta_i^c - \delta_i}{\delta_i^c - \delta_i^0} \tag{10.11}$$

which substituted into (10.10) yields

$$D_i = \frac{\delta_i^c(\delta_i - \delta_i^0)}{\delta_i(\delta_i^c - \delta_i^0)} \tag{10.12}$$

In this way, the damage variables D_i are calculated as a function of the relative separation between the faces of the laminate δ_i, which is provided by the FE solution and the values δ_i^0, δ_i^c calculated previously.

In summary, there are four distinct stages in the material behavior described by Fig. 10.4:

- linear elastic undamaged material behavior (line OA), with the associated constitutive equation (10.4).
- damage initiation (point A), with the associated criterion (10.6).
- damage evolution (line AC), with the associated damage evolution equation (10.5), (10.12).
- fracture (crack formation), with the associated crack formation criterion (10.2).

As previously stated, the formulation presented in this section works only for pure mode I, or II, or III. The general case of the mixed mode loading is presented in the following section.

10.1.2 Mixed Mode Cohesive Model

When the interface of the laminated material is under mixed modes, all three traction components $\sigma_I, \sigma_{II}, \sigma_{III}$ and all three separation components $\delta_I, \delta_{II}, \delta_{III}$ are active. In other words, mixed mode implies that two or more pairs (σ_i, δ_i) are non-vanishing, with $i = I, II, III$. However, to reduce the burden of experimentation, it is assumed that the penalty stiffness (10.7) is the same for all modes $\widetilde{K} = \widetilde{K}_i$.

Mixed mode ratios can be defined between pairs of mode components. For example, in terms of separations

$$\beta_{\delta_{II}} = \frac{\delta_{II}}{\delta_I} \quad ; \quad \beta_{\delta_{III}} = \frac{\delta_{III}}{\delta_I} \tag{10.13}$$

or in terms of ERR

$$\beta_{G_{II}} = \frac{G_{II}}{\sum_1^3 G_i} \quad ; \quad \beta_{G_{III}} = \frac{G_{III}}{\sum_1^3 G_i} \tag{10.14}$$

Regardless of the definition used, mixed mode ratios are just parameters characterizing the mixed mode state, which allow for a simplification of the analysis by assuming that decohesion progresses at constant mixed mode ratios. It is further assumed that the modes are uncoupled even though they occur simultaneously. That is, the stress-separation relationship for each uncoupled mode is again expressed by (10.4), separately for each mode.

Next, a mixed mode separation is defined by the L^2 norm of the mode separations, i.e.,

$$\delta_m = \sqrt{\sum_{i=1}^{M} \delta_i^2} \tag{10.15}$$

where M is the number of modes involved (i.e., 2 or 3 modes). Next, the single mode damage initiation criterion in (10.6) is replaced, for example, by a quadratic stress criterion

$$\sum_{i=1}^{M} \left(\frac{\sigma_i}{\sigma_i^0}\right)^2 = 1 \tag{10.16}$$

Other mixed mode damage initiation criteria can be selected in *Abaqus*, such as the *quadratic displacement criterion*, or a *maximum-stress/maximum-displacement criterion.* Regardless of the damage initiation criterion selected, the goal is to calculate the mixed mode separation δ_m^0 at damage onset under mixed mode loading.

For the case of mode I and II only, $M = 2$ and the *equivalent mixed mode separation at damage onset* δ_m^0 is found as follows. First, rewrite the damage initiation criteria (10.16) in terms of separations using (10.7) and (10.4)-(10.5), taking into account that $D_i = 0$ up to damage initiation. Therefore, mixed mode damage initiation is predicted by

$$\left(\frac{\delta_I}{\delta_I^0}\right)^2 + \left(\frac{\delta_{II}}{\delta_{II}^0}\right)^2 = 1 \tag{10.17}$$

Next, rewrite (10.15) using the first of (10.13) to get

$$\delta_I = \frac{\delta_m}{\sqrt{1+\beta^2}} \tag{10.18}$$

and using (10.13) again

$$\delta_{II} = \beta\frac{\delta_m}{\sqrt{1+\beta^2}} \tag{10.19}$$

Now, substitute (10.18) and (10.19) into (10.17), taking into account that since (10.17) represents damage initiation, one should write δ_m^0 for δ_m. Therefore,

$$\delta_m^0 = \sqrt{(\delta_I^0)^2(\delta_{II}^0)^2\frac{1+\beta^2}{(\delta_{II}^0)^2+\beta^2(\delta_I^0)^2}} \tag{10.20}$$

The quantities δ_i^0 in (10.20) represent the separations at damage onset during single mode loading, calculated with (10.7), and β is the mixed mode ratio, which is assumed to be constant during the damage process.

A mixed mode crack propagation criterion is now needed to replace the single mode criterion (10.2). A possible choice is to use an ERR power criterion as follows

$$\sum_{i=1}^{3}\left(\frac{G_i}{G_{ic}}\right)^{\alpha_i} = 1 \tag{10.21}$$

which attempts to predict fracture under mixed mode conditions, similarly to point C in Fig. 10.4 for the single mode situation. To reduce the burden of experimentation, it is customary to assume that the exponents are the same for all modes, i.e., $\alpha_i = \alpha$.

Each of the single mode component G_i can be calculated by one of two methodologies. Some authors [176, 178] calculate each single mode component ERR by considering the area OABB' in Fig. 10.4, thus including the recoverable energy OBB' in the definition of G_i. This approach is indirectly related to linear elastic fracture mechanics (LEFM). Other authors [188] use a damage mechanics approach where each single mode component ERR is calculated by considering only the unrecoverable energy represented by the area OAB. Both approaches lead to the same results for a single-mode delamination since loss of adhesion occurs at δ_i^c where both approaches predict the same values for G_i (point B reaches point C). However, different results are obtained for mixed-mode delamination because the crack propagates when an interaction criterion is satisfied, the latter involving the ratios G_i/G_{ic}.

In principle, the approach based on LEFM should lead to conservative predictions of the load carrying capacity for mixed-mode delamination. On the other hand, since the total energy dissipated during the delamination at each point is not released instantaneously as assumed in LEFM, the damage mechanics based definition appears appropriate especially when the size of the nonlinear fracture process zone ahead of the delamination front is not negligible as it may occur in the case of laminated composite materials where the damage zone may be comparable to or larger than the single-ply thickness, which generally scales with the near-tip stress field.

In the sequel, the damage mechanics approach is used. That is, for each single mode separation δ_i, the single mode component ERR G_i is calculated as the area OAB in Fig. 10.4, which represents dissipated energy, i.e.,

$$A_{OAB} = A_{OAF} - A_{OBF} \tag{10.22}$$

where A_{OAF} is the single mode critical ERR G_{ic}, and A_{OBF} can be calculated based on the geometry in Fig. 10.4 as

$$A_{OBF} = \frac{1}{2} BB' \times OF = \frac{1}{2}\ K_i \delta_i^0 \frac{\delta_i^c - \delta_i}{\delta_i^c - \delta_i^0}\ \delta_i^c \tag{10.23}$$

where $\delta_i^c = OF$, $k_i \delta_i^0 = \sigma_i^0$, and σ_i/σ_i^0 is given by (10.11). Based on (10.22), (10.23), the single mode component ERR at the moment of mixed mode fracture is calculated as

$$G_i = G_{ic} - \frac{1}{2} K_i \delta_i^0 \delta_i^c \frac{\delta_i^c - \delta_i}{\delta_i^c - \delta_i^0} \tag{10.24}$$

The assumed mode decomposition (10.24) is necessary so that each single mode component ERR G_i corresponding to mixed mode fracture can be expressed as a function of the single mode separation δ_i, i.e., $G_i = G_i(\delta_i)$. All other quantities in (10.24) are known, as follows:

- G_{ic} is the single mode critical ERR (material parameter).
- δ_i^0 is the separation at damage onset under single mode loading (point A in Fig. 10.4), given by (10.7)
- δ_i^c is the separation at fracture under single mode loading (point C in Fig. 10.4), given by (10.9).

The single mode ERR components of the mixed mode condition in (10.24) have to satisfy the energy criterion (10.21) at the moment of fracture (crack propagation). For the case of two modes, and assuming $\alpha_i = \alpha = 2$ in (10.21), the mixed mode separation at fracture (point C) is calculated in [188, (15)] as

$$
\begin{aligned}
\delta_m^c =& \frac{\sqrt{1+\beta}}{\beta^2(\delta_I^{0F})^2 + (\delta_{II}^{0F})^2} \times \\
& \{\delta_I^0(\delta_{II}^{0F})^2 + \beta\delta_{II}^0(\delta_I^{0F})^2 \\
& + \delta_I^{0F}\delta_{II}^{0F}\sqrt{(\delta_{II}^{0F})^2 - (\delta_{II}^0)^2 + 2\beta\delta_I^0\delta_{II}^0 - \beta^2(\delta_I^0)^2 + \beta^2(\delta_I^{0F})^2}\Big\}
\end{aligned}
\qquad (10.25)
$$

where $\delta_I^{0F} = \delta_I^c - \delta_I^0, \delta_{II}^{0F} = \delta_{II}^c - \delta_{II}^0$.

There are three energy crack extension criteria implemented in *Abaqus*:

– the power equation criterion in (10.21).

– the BK criterion, which is another closed form function of (G_i, G_{ic}).

– a user input of the total mixed mode ERR at fracture $G_c = \sum_1^3 G_{ic}$.

Once the mixed mode separation at fracture (10.25) is calculated based on the selected criterion, and the mixed mode separation at damage onset is known based on (10.20), the damage variable for mixed mode conditions can be expressed in a similar manner as for the case of single mode condition (10.12) by satisfying the requirement in (10.8), i.e.,

$$
D_m = \frac{\delta_m^c(\delta_m - \delta_m^0)}{\delta_m(\delta_m^c - \delta_m^0)} \qquad (10.26)
$$

where the onset and fracture separations δ_m^0 and δ_m^c are calculated based on (10.20) and (10.25), respectively; and δ_m is the current level of separation under mixed mode conditions obtained using (10.15) in terms of the single modes separations δ_i provided by the FE model. The stiffness degradation for the cohesive material is then calculated according with (10.5). The stress softening evolution of the cohesive material under mixed mode conditions is similar to the one depicted in Fig. 10.4, by substituting δ_m^0, δ_m^c for δ^0, δ^c.

Besides the energy approach to mixed mode damage evolution as presented here, where the crack extension criterion is a function of ERR, and δ_m^c is calculated based on this energy criterion, *Abaqus* also has implemented a displacement approach to damage evolution. The displacement approach requires the user to directly provide the mixed mode separation at the moment of fracture δ_m^c, which is then directly used in (10.26).

Example 10.1 *A laminated double cantilever beam (DCB)* 100 *mm long and* 20 *mm wide is made up of two layers bonded by an adhesive layer of negligible thickness. Each layer is* 1.5 *mm thick. Apply a loading system to induce delamination and mode I crack growth through the adhesive layer. Assuming linear elastic behavior, create a 2D model of the DCB using CZM elements to represent the adhesive layer. Use the* `element-based` *cohesive behavior implemented in* Abaqus *to represent the adhesive.*

Solution to Example 10.1 *To facilitate the solution of this example, run the Python script* `ws_composites_dcb.py` *in your* `Work Directory`*. The script can be fetched from the examples in the* Abaqus *documentation as follows*

```
# open a Windows command prompt via the Windows Taskbar and Start Menu
Start, Run, Open [cmd], OK
# Navigate to the folder C:\SIMULIA\User\Ex_10.1
# you may need to create the directory using the command: mkdir Ex_10.1
# type the following command and press the Enter-key
abaqus fetch job=ws_composites_dcb.py
# this will copy the file ws_composites_dcb.py into the Work Directory
```

The script creates two models, of which only the 2D model named `coh-els` *is used in this example. The model contains the parts, assembly, materials, sections, sets, and surfaces needed to apply boundary conditions, constraints, and to define the output requests fields.*

During model set up, the cohesive elements representing the adhesive layer are defined with a finite thickness. This is done only to facilitate manipulation and the definition of constraints. At the end of the setup process, the thickness of the cohesive elements is adjusted to zero, as explained in Section v, p. 367. That is, cohesive elements COH2D4 with zero thickness are used in this example.

Although the loading conditions applied in this example induce crack propagation in mode I only, Abaqus *requires defining the parameters corresponding to all three propagation modes.*

```
Menu:   File, Set Work Directory, [C:\SIMULIA\User\Ex_10.1], OK
```

i. Running the Python script

```
Menu:   File, Run Script, [ws_composites_dcb.py], OK
    # notice that the model has been named dcb.cae
Menu:   Save As, [Ex_10.1.cae], OK
```

ii. Defining materials, sections, and assigning sections to parts.

The `coh-els` *model is set up with units* $[m, Pa]$*. A material named 'bulk' contains the undamaged material properties of the laminas:* $E_1 = 135.3 \times 10^9\ Pa; E_2 = E3 = 9 \times 10^9\ Pa; \nu_{12} = \nu_{13} = 0.24; \nu_{23} = 0.46; G_{12} = G_{23} = 4.5 \times 10^9\ Pa; G_{13} = 3.3 \times 10^9\ Pa.$

To enter the properties of the adhesive, a new material called 'adhesive' must be defined using the pseudo code below. The adhesive stiffness values (10.4) *are:* $K_I = E/K_{nn} = 570 \times 10^{12}\ Pa$, $K_{II} = E/K_{ss} = 570 \times 10^{12}\ Pa$, *and* $K_{III} = E/K_{tt} = 570 \times 10^{12}\ Pa$, *where* K_{nn}, K_{ss}, K_{tt} *are penalty parameters that are mentioned here only because that is how they are labeled in* Abaqus*/CAE. The adhesive strength values* (10.6) *are:* $\sigma_I^0 = 5.7E7$, $\sigma_{II}^0 = 5.7E7$, $\sigma_{III}^0 = 5.7E7$*. The adhesive parameters for the Benzeggagh-Kenane (BK) crack extension criterion are: Power* 2.284*, and fracture energies* $G_{Ic} = 280\ J/m^2$, $G_{IIc} = 280\ J/m^2$, $G_{IIIc} = 280\ J/m^2$ *(see* (10.2)*).*

```
Module: Property
Model:  coh-els
Part:   adhesive
Menu:   Material, Manager
    Create, Name [cohesive], Mechanical, Elasticity, Elastic
```

```
        Type: Traction, E/Knn [5.7E14], G1/Ktt [5.7E14], G2/Kss [5.7E14]
        # damage onset criterion
        Mechanical, Damage for Traction Separation Laws, Quads Damage
        Nominal Stress Normal-only Mode [5.7E7]
        Nominal Stress First Direction  [5.7E7]
        Nominal Stress Second Direction [5.7E7]
        # crack extension criterion and required parameters
        # Benzeggagh-Kenane (BK) is used, see Abaqus documentation
        Suboptions, Damage Evolution
        Type: Energy, Mixed mode behavior: BK, # checkmark: Power [2.284]
        Normal Mode Fracture Energy [280] # GIc
        Shear Mode Fracture Energy First Direction  [280] # GIIc
        Shear Mode Fracture Energy Second Direction [280] # GIIIc
        OK, OK, # close Material Manager pop-up window
    Menu:   Section, Manager
        # a section named 'bulk' defines the laminas
        # create a section for the adhesive
        Create, Name [cohesive], Category: Other, Type: Cohesive, Cont
        Material: cohesive, Response: Traction Separation
        # checkmark: Out-of-plane thickness [0.02], OK
        # close Section Manager pop-up window
    Menu:   Assign, Section
        # pick the part, Done, Section: cohesive, OK
```

iii. *Defining the analysis steps*

```
    Module: Step
    Menu:   Step, Create, Cont
        # geometrical non-linearity for large displacements
        Tab: Basic, Nlgeom: On
        # parameters to setup the time increments for the solution
        Tab: Incrementation, Maximum number of increments [1000]
        Increment size: Initial [0.01], Minimum [1.0E-8], OK
    Menu:   Output, Field Output Requests, Edit, F-Output-1
        Output Variables: # expand: State/Field/User/Time
        # checkmark: STATUS, OK
        # the STATUS variable is an on/off switch that indicates the CAE
        # postprocessor when to stop drawing the cohesive elements
        # a completely degraded (damaged) element will have STATUS 0 (off)
    Menu:   Output, History Output Requests, Create
        Name [H-Output-2], Step: Step-1, Cont
        Domain: Set : top
        # expand: Displ/Veloc/Accel, # expand: U,  # checkmark: U2
        # expand: Forces/Reactions,  # expand: RF, # checkmark: RF2, OK
        # the values of U2 (imposed displacement) and RF2 (reaction force)
        # at the specified set (node) will be stored for postprocessing
```

iv. *Meshing the model*

The sweep technique is used because the length of the cohesive elements is orientation-dependent. For this case they need to be defined from the bottom to the top surface of the adhesive layer because their elongation will represent the separation of the laminas.

A viscosity parameter is used along with the COH2D4 element to speed up the convergence of the iterative solution. You may change its value to see how it affects the solution.

```
Module: Mesh
Model:  coh-els
Object: Part: beam
Menu:   Mesh, Controls, Element Shape: Quad, Technique: Structured, OK
Menu:   Mesh, Element Type
    Family: Plane Strain, # checkmark: Incompatible modes
    # Notice element CPE4I has been assigned, OK
Menu:   Seed, Edges
    # pick the top horizontal line, Done
    Method: By number, Number of elements [400], OK
    # pick the left vertical line, Done
    Method: By number, Number of elements [2], OK
Menu:   Mesh, Part, Yes
Object: Part: adhesive
Menu:   Mesh, Controls, Element Shape: Quad, Technique: Sweep
    Redefine Sweep Path
    # if the highlighted path coincides with the thickness direction
    Accept Highlighted
    # otherwise change accordingly to accomplish that, OK
Menu:   Mesh, Element Type
    Family: Cohesive, Viscosity: Specify [1.0E-5]
    # notice element COH2D4 has been assigned, OK
Menu:   Seed, Edges
    # pick the top horizontal line, Done
    Method: By number, Number of elements [280], OK
    # pick the left vertical line, Done
    Method: By number, Number of elements [1], OK
Menu:   Mesh, Part, Yes
```

v. Creating model constraints.
The use of a finite-thickness part to represent the adhesive layer facilitated the definition of the cohesive elements properties. The constraints applied here guarantee that the top and bottom surfaces of the cohesive elements are coincident once the solution process starts, effectively defining the initial length of the cohesive elements as zero. Although it is not visually shown, the procedure below adjusts the surfaces defined as slaves to follow those defined as masters.

```
Module: Interaction
Menu:   Constraint, Create, Name [top], Type: Tie, Cont
    Choose the master type: Surface, Surfaces, top, Cont
    Choose the slave type:  Surface, coh-top, Cont
    Position Tolerance: Specify distance [0.002], OK
Menu:   Constraint, Create, Name [bot], Type: Tie, Cont
    Choose the master type: Surface, bot, Cont
    Choose the slave type:  Surface, coh-bot, Cont
    Position Tolerance: Specify distance [0.002], OK
```

vi. Applying BC and loads

```
Module: Load
Menu:   BC, Create, Name [top], Step: Step-1, Disp/Rota, Cont
    Sets, top, Cont, # checkmark: U1 [0], # checkmark: U2 [0.006], OK
Menu:   BC, Create, Name [bot], Step: Step-1, Disp/Rota, Cont
    Sets, bot, Cont, # checkmark: U1 [0], # checkmark: U2 [-0.006], OK
    # due to the symmetry of the model no additional BCs are needed
```

vii. *Solving and visualizing the results*

```
Module: Job
Menu:   Job, Manager
    Create, Name [Ex-10-1], Cont, OK, Submit, # when completed, Results
Module: Visualization
Menu:   Plot, Deformed Shape # notice the beams are now separated
Menu:   Plot, Contours, On Deformed Shape
Menu:   Animate, Time History # see the progression of the delamination
```

To visualize the force–separation behavior in the model:

```
Module: Visualization
Menu:   Result, History Output
  Output Variables: Reaction force: RF2 PI, Save As [coh-els-RF2], OK
  Output Variables: Spatial displ:  U2 PI,  Save As [coh-els-U2],  OK
  # close the History Output pop-up window
Menu:   Tools, XY Data, Create, Source: Operate on XY data, Cont
  Operators: Combine(X,X), # pick: coh-els-U2, Add to Expression
  # pick: coh-els-RF2, Add to Expression,
      Save As [cohesive elements], OK
  # close Operate on XY Data pop-up window
Menu:   Tools, XY Data, Plot, cohesive elements
```

The resulting plot should look like the dash line in Fig. 10.6. The drop of the reaction force occurs when the cohesive elements start to degrade (damage). As the cohesive elements degrade the reaction force reduces continuously. In the visualization module, zoom in close to the crack tip and identify the cohesive elements representing the adhesive layer.

Example 10.2 *Solve Example 10.1 using surface-based cohesive behavior instead of element-based cohesive method.*

Solution to Example 10.2 *This example is solved modifying the* `.cae` *file defined in Example 10.1. The part defined in Example 10.1 representing the adhesive layer is removed, then redefined as an interaction between the coincident surfaces of the two laminas. That is, element-based cohesive method features are replaced by surface-based cohesive method features.*

```
Menu: File, Set Work Directory, [C:\SIMULIA\User\Ex_10.2], OK
Menu: File, Open, [C:\SIMULIA\User\Ex_10.1\Ex_10.1.cae], OK
# on the Model-tree
# right-click: coh-els, Rename, [coh-surf], OK
Menu: File, Save As, [C:\SIMULIA\User\Ex_10.2\Ex_10.2.cae], OK
```

i. *Deleting unnecessary features*

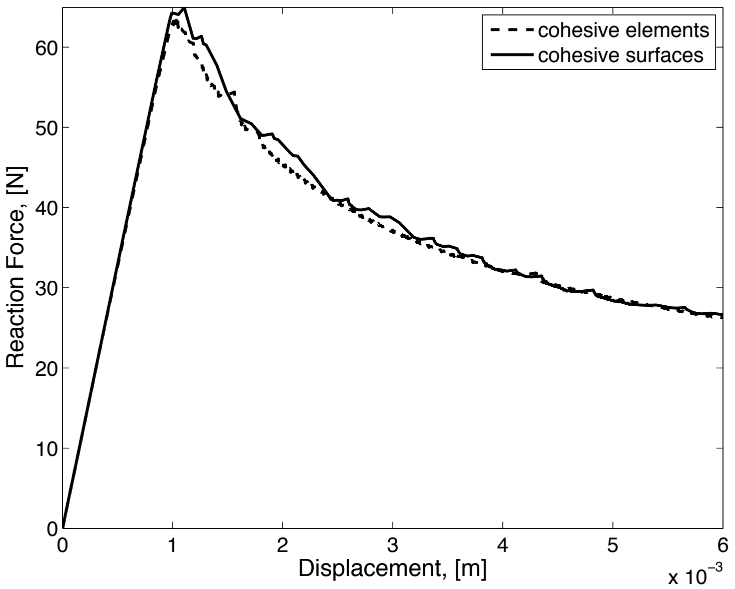


Fig. 10.6: Comparison of force–separation behavior for models using cohesive elements (Example 10.1) and cohesive surfaces (Example 10.2).

```
Module: Assembly
Model:  coh-surf
Menu:   Tools, Surface, Manager
    # pick: coh-top, Delete, Yes
    # pick: coh-bot, Delete, Yes, # close Surface Manager
Module: Interaction
Menu:   Constraint, Manager
    # pick: bot, Delete, Yes, # pick: top, Delete, Yes
    # close Constraint Manager
    # on the model-tree
    # expand: coh-surf, # expand: Assembly, # expand: Instances
    # right-click: adhesive-1, Delete, Yes
Menu:   Feature, Regenerate
Module: Mesh
Model:  coh-surf
Object: Part:   beam
Menu:   Mesh, Delete Part Mesh, Yes
```

ii. *Creating a partition.*
At this point the model consists only of two stacked beams. Now, it is necessary to divide the interface between the beams to represent the region where the adhesive is applied. A partition will be created.

```
Menu:   Tools, Partition
    Type: Edge, Method: Enter parameter
    # pick bottom edge of beam, Done, Normalized edge parameter [0.7]
    Create Partition, Done, # close the Create Partition pop-up window
```

iii. *Meshing the part*

```
Menu:   Mesh, Part, Yes
```

iv. *Creating sets and surfaces.*
The surface called top *in Example 10.1 will be redefined to be the whole interface between the beams. Since the two beams are stacked, selecting only one of the coincident surfaces is cumbersome. To facilitate selecting the desired surface, a useful visualization tool is introduced –Display Group Toolbar.*

```
Module: Assembly
    # locate the Toolbar named Display Group, If it is not available
    # enable it using Menu: View, Toolbars, Display Group
    # click the first icon of Display Group Toolbar (Replace Selected)
    # pick the top beam, Done, # only the picked part will be visible
Menu:   Tools, Surface, Edit, top
    # pick the two lines forming the bottom edge of the beam, Done
Menu:   Tools, Set, Create
    Name [bond], Type: Geometry, Cont
    # pick the longest line at the bottom of the beam, Done
    # click on third icon of Display Group Toolbar (Replace All)
    # the full assembly will be visualized
```

The set named bond *will represent the adhesive layer in the same way the part named* adhesive-1 *did in Example 10.1.*

v. *Defining interactions to represent the adhesive layer.*
First the properties and characteristics of the contact are defined. Then the interacting surfaces are selected.

```
Module: Interaction
Menu:   Interaction, Property, Create
  Name [coh], Type: Contact, Cont
  Mechanical, Tangential Behavior, Friction formulation: Frictionless
  Mechanical, Normal Behavior, Pressure-Overclosure: Hard Contact
  Constraint enforcement method: Default
  Mechanical, Cohesive Behavior
  Eligible Slave Nodes: Specify the bounding node set in surf-to-surf
  Traction-separation Behavior: Specify stiffness coeffs, Uncoupled
  Knn [5.7e14], Kss [5.7e14], Ktt [5.7e14] # KI, KII, and KIII
  Mechanical, Geometric Properties,
      Out-of-plane surface thickness [0.02]
  Mechanical, Damage, Criterion: Quadratic traction
  Normal Only [5.7e7], Shear-1 Only [5.7e7], Shear-2 [5.7e7]
  # these define the onset of damage in the adhesive layer
  # checkmark: Specify damage evolution
  Type: Energy,
      # checkmark: Specify mixed mode behavior: Benzeggagh-Kenane
  # checkmark: Specify power-law/BK exponent [2.284]
  Normal Fracture Energy [280], # Critical crack propagation GIc
  1st Shear Fracture Energy [280] # GIIc
  2nd Shear Fracture Energy [280] # GIIIc
  # checkmark: Specify damage stabilization, Viscosity coeff. [1e-5]
```

```
    # it plays the same role as the one used for cohesive elements
    OK # the properties of the adhesive layer have been defined
  Menu:   Interaction, Create
    Name [coh], Step: Initial, Type: Surface-to-surface (Standard), Cont
    Surfaces, # pick: bot, Cont # this is the master surface
    Surfaces, # pick: top, Cont # this is the slave surface
    Discretization method: Node to surface
    Tab: Bonding
    # checkmark: Limit bonding to slave nodes in subset: bond
    # This represents the adhesive layer bonding the beams, OK
```

vi. *Solving and visualizing the results*

```
  Module: Job
  Menu:   Job, Manager
    Create, Name [Ex-10-2], Cont, OK, Submit, # when completed, Results
  Module: Visualization
  Menu:   Plot, Deformed Shape # notice the beams have been separated
  Menu:   Plot, Contours, On Deformed Shape
  Menu:   Animate, Time History # progression of the delamination
```

To visualize the force–separation behavior in the model:

```
  Module: Visualization
  Menu:   Result, History Output
    Output Variables: Reaction force: RF2 PI,
        Save As [coh-surf-RF2], OK
    Output Variables: Spatial displ:  U2 PI,
        Save As [coh-surf-U2],  OK
    # close the History Output pop-up window
  Menu:   Tools, XY Data, Create, Source: Operate on XY data, Cont
    Operators: Combine(X,X), # pick: coh-surf-U2, Add to Expression
    # pick: coh-surf-RF2, Add to Expression,
        Save As [cohesive surfaces], OK
    # close Operate on XY Data pop-up window
  Menu:   Tools, XY Data, Plot, cohesive surfaces
```

The resulting plot should look like the solid line in Fig. 10.6. The drop-off of the reaction force occurs when the separation between the surfaces reaches a critical value, δ_i^0, which correspond to a critical stress value, σ_i^0. Unlike Example 10.1, where cohesive elements are used, in this example the progressive damage of the interphase is calculated by Abaqus *without the use of particular elements. In the visualization module, zoom-in close to the crack-tip and notice that no elements are depicted to represent the adhesive layer.*

10.2 Virtual Crack Closure Technique

The virtual crack closure technique (VCCT) can be used to analyze delaminations in laminated materials using a fracture mechanics approach. The method implements Linear Elastic Fracture Mechanics (LEFM). Only brittle crack propagation is modeled. The energy dissipated by the formation of plastic zones at the crack-tip is not considered.

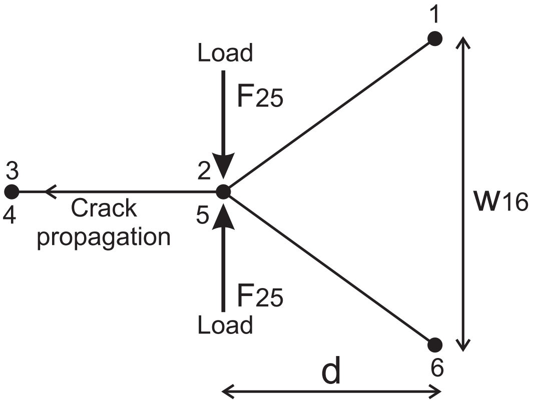


Fig. 10.7: VCCT

The condition for crack propagation is based on Griffith's principle (10.2). For the case of single mode deformation under mode I conditions, the crack grows when

$$\frac{G_I}{G_{Ic}} \geq 1 \tag{10.27}$$

where G_I is the ERR for mode I crack formation and G_{Ic} a material property representing the critical ERR for mode I crack formation.

The definition of ERR is given by (10.1). In VCCT, the Irwin principle (10.3) is used to calculate the change in strain energy ΔU, which is considered to be equal to the work required for crack closure $W_{closure}$.

By substituting (10.3) and (10.1) into (10.27), the condition for crack propagation under mode I loading becomes

$$\frac{W_{closure}/\Delta A}{G_{Ic}} \geq 1 \tag{10.28}$$

Abaqus [189] calculates $W_{closure}$ from the FE nodal displacements and forces as illustrated in Fig. 10.7. Initially the crack surfaces are rigidly bonded. The nodal forces at the coincident nodes $2-5$ are calculated from the FE solution. The hypothesis of *self-similar crack propagation* is used, which says that during crack propagation, the crack configuration between nodes $2-3-4-5$ will be similar to the crack configuration between nodes $1-2-5-6$. This implies that the separation between nodes $2-5$ after crack propagation will be equal to the separation between nodes $1-6$ before crack propagation: $v_{2,5} = v_{1,6}$. If the nodes $2-5$ open (crack propagation), the elastic work required to close the crack is

$$W_{closure} = \frac{1}{2} F_{2,5}\ v_{2,5} = \frac{1}{2} F_{2,5}\ v_{1,6} \tag{10.29}$$

By substituting (10.29) in (10.28), the condition for crack propagation becomes

$$\frac{F_{2,5}\ v_{1,6}}{2\Delta A} \frac{1}{G_{Ic}} \geq 1 \tag{10.30}$$

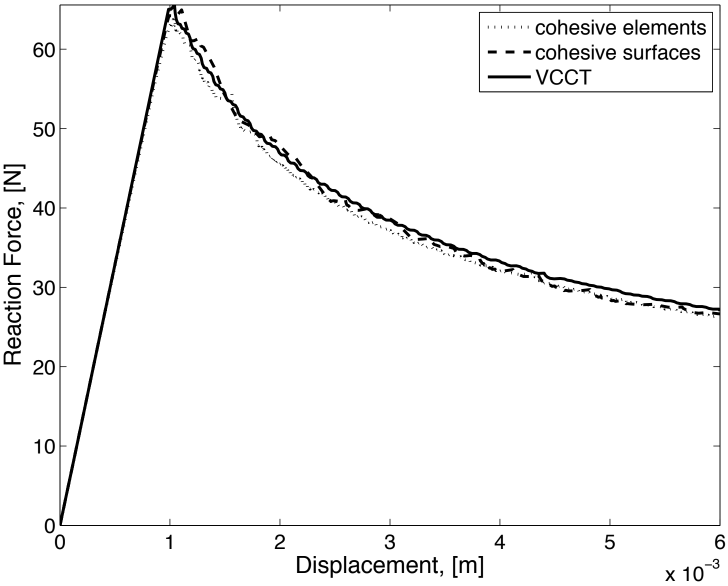


Fig. 10.8: Comparison of force–separation behavior using Virtual Crack Closure Technique (VCCT), cohesive elements, and cohesive surfaces.

The area ΔA of the newly formed crack is $\Delta A = d \times b$, where d is the length of the finite element undergoing crack propagation, and b is the width of the crack.

The VCCT method works similarly for the modes II or III, by considering the corresponding components of the separation and nodal forces. A refined formulation called Jacobian derivative method is available in [190].

The crack propagation criterion (10.30) applies for single mode loading only, as it is derived from the single mode criterion in (10.2). For mixed mode loading, the single mode crack propagation criterion (10.2) has to be replaced with a mixed mode criterion. For example, one could use the power equation (10.21), where the critical ERRs G_{ic}, with $i = I, II, III$, are material properties and the ERRs G_i are calculated similarly to (10.30) by using VCCT. Alternatively, one could use the BK equation [191] or the Reeder equation [180,192].

Example 10.3 *Solve Example 10.1 (p. 364) using the virtual crack closure technique.*

Solution to Example 10.3 *This example is solved modifying the* `cae` *file defined in Example 10.2, p. 368. Initially, the contact definition must be changed deleting the properties associated with the surface-based cohesive formulation and redefining the contact characteristics. Then, the Output Requests must be modified to include the calculation of Strain Energy Release Rate, G, in the model. And finally, the Keyword Editor will be used to add the VCCT features before submitting the model to the solver.*

```
Menu: File, Set Work Directory, [C:\SIMULIA\User\Ex_10.3], OK
Menu: File, Open, [C:\SIMULIA\User\Ex_10.2\Ex_10.2.cae], OK
# on the Model-tree
# right-click: coh-surf, Rename, [vcct], OK
Menu: File, Save As, [C:\SIMULIA\User\Ex_10.3\Ex_10.3.cae], OK
```

i. *Deleting unnecessary features*

```
Module: Interaction
Menu:   Interaction, Property, Edit, coh
    # pick: Cohesive Behavior, Delete
    # pick: Damage, Delete, OK
```

ii. *Redefining the contact definition*

```
Menu:   Interaction, Edit, coh
    Sliding formulation: Small sliding
    Tab: Clearance
    Initial clearance: Uniform value across slave surface,
        [1.0e-7], OK
```

The use of an initial clearance will facilitate the solution of the model by preventing initial overclosures product of the meshing algorithm. See Abaqus *documentation for details.*

iii. *Modifying Outfield Requests*

```
Module: Step
Step:   Step-1
Menu:   Output, Field Output Requests, Edit, F-Output-1
    Output Variables: # expand: Failure/Fracture
    # checkmark: ENRRT, # checkmark: BDSTAT, OK
    # ENRRT and BDSTAT allow tracking damage evolution
```

iv. *Adding Keywords related to VCCT*

```
Menu:   Model, Edit Keywords, vcct
    # click on the line that starts with the text: *Contact Pair
    Add After, # Type in the following text:
    [*Initial Conditions, type=CONTACT
    top, bot, bond]
    # click on the line that starts with the text: *Static
    Add After, # Type in the following text:
    [*Debond, slave=top, master=bot
    *Fracture criterion, type=VCCT, mixed mode behavior=BK,
        tolerance=0.1
    280.0,280.0,280.0,2.284]
    OK
```

v. *Solving and visualizing the results*

```
Module: Job
Menu:   Job, Manager
  Create, Name [Ex-10-3], Cont, OK, Submit, # when completed, Results
Module: Visualization
Menu:   Plot, Deformed Shape # notice the beams have been separated
Menu:   Plot, Contours, On Deformed Shape
Menu:   Animate, Time History # see progression of the delamination
```

To visualize the force separation behavior in the model:

```
Module: Visualization
Menu:   Result, History Output
    Output Variables: Reaction force: RF2 PI, Save As [vcct-RF2], OK
    Output Variables: Spatial displ:  U2 PI,  Save As [vcct-U2],  OK
    # close the History Output pop-up window
Menu:   Tools, XY Data, Create, Source: Operate on XY data, Cont
    Operators: Combine(X,X), # pick: vcct-U2, Add to Expression
    # pick: vcct-RF2, Add to Expression, Save As [vcct], OK
    # close Operate on XY Data pop-up window
Menu:   Tools, XY Data, Plot, vcct
```

The resulting plot should look like the solid line in Fig. 10.8. The variable `BDSTAT` *stores the damage evolution associated with the bond between two nodes that were made coincident during the setup of the model. Its value ranges from 0, fully damaged bond, to 1, undamaged bond.* `BDSTAT` *can be used to find the location of the crack-tip at any stage of the delamination. In the* `Visualization Module`*, visualize* `BDSTAT` *and zoom-in to locate the crack-tip.*

Suggested Problems

Problem 10.1 *Retrieve the values of reaction force vs. displacement from* Abaqus *for Examples 10.1 and 10.2, pp. 364, 368. Plot them together and discuss the difference. Your plot should be similar to the one shown in Fig. 10.6. To retrieve the data as a text file proceed as follows:*

```
Module: Visualization
Menu:   Report, XY
    # pick: cohesive surfaces, # cohesive elements for Example 10.1
    Tab: Setup, Name [coh_surf.rpt], # coh_ele.rpt for Example 10.1
    # un-checkmark: Append to file, OK
    # the report file will be saved in your Work Directory
```

Problem 10.2 *Retrieve the values of reaction force vs. displacement for Example 10.3 (p. 10.3) and plot them along those of Examples 10.1 and 10.2, p. 364, 368. Discuss the differences. The comparative plot should be similar to the one shown in Fig. 10.8.*

Appendix A

Tensor Algebra

Tensor operations are needed for the derivation of some of the equations in this textbook. Since most of these operations are not easily found in textbooks, they are presented here for reference [193].

A.1 Principal Directions of Stress and Strain

Since stress and strain tensors are symmetric and of second order, they have three real principal values and three orthogonal principal directions. The principal values λ^q and directions n_i^q of the stress tensor σ_{ij} satisfy the following

$$[\sigma_{ij} - \lambda^q \delta_{ij}] n_i^q = 0 \tag{A.1}$$

$$n_i^q n_j^q = 1 \tag{A.2}$$

where δ_{ij} is the Kronecker delta ($\delta_{ij} = 1$ if $i = j$, zero otherwise). Each of the principal directions is described by its direction cosines with respect to the original coordinate system.

The principal directions are arranged by rows into a matrix $[A]$. Then, the diagonal matrix $[A^*]$ of the principal values is

$$[A^*] = [a][A][a]^T \tag{A.3}$$

It can be shown that $[a]^{-1} = [a]^T$, where $[a]$ is the transformation matrix given by (1.21).

A.2 Tensor Symmetry

Minor symmetry provides justification for using contracted notation (Section 1.5). Minor symmetry refers to identical values of tensor components when adjacent subscripts are swapped. For example, minor symmetry of the stiffness tensor **C** means

$$A_{ijkl} = A_{jilk} = A_{\alpha\beta} \tag{A.4}$$

Major symmetry refers to identical values when adjacent pairs of subscripts are swapped, or when contracted subscripts are swapped. For example,

$$\begin{aligned} A_{ijkl} &= A_{klij} \\ A_{\alpha\beta} &= A_{\beta\alpha} \end{aligned} \tag{A.5}$$

A.3 Matrix Representation of a Tensor

A tensor A_{ijkl} with a minor symmetry has only 36 independent constants. Therefore it can be represented in contracted notation by a 6 × 6 matrix. Let $[a]$ be the contracted form of the tensor A. Each element of $[a]$ corresponds to an element in the tensor A according to the following transformation

$$a_{\alpha\beta} = A_{ijkl} \tag{A.6}$$

with

$$\begin{aligned} \alpha &= i \text{ when } i = j \\ \alpha &= 9 - (i + j) \text{ when } i \neq j \end{aligned} \tag{A.7}$$

The same transformations apply between β and k and l, or in matrix representation, as

$$[a] = \begin{bmatrix} A_{1111} & A_{1122} & A_{1133} & A_{1123} & A_{1113} & A_{1112} \\ A_{2211} & A_{2222} & A_{2233} & A_{2223} & A_{2213} & A_{2212} \\ A_{3311} & A_{3322} & A_{3333} & A_{3323} & A_{3313} & A_{3312} \\ A_{2311} & A_{2322} & A_{2333} & A_{2323} & A_{2313} & A_{2312} \\ A_{1311} & A_{1322} & A_{1333} & A_{1323} & A_{1313} & A_{1312} \\ A_{1211} & A_{1222} & A_{1233} & A_{1223} & A_{1213} & A_{1212} \end{bmatrix} \tag{A.8}$$

It is convenient to perform tensor operations using contracted form, especially if the result can be represented also in contracted form. This saves memory and time since it is faster to operate on 36 elements than on 81 elements. Examples of these operations are the inner product of two fourth-order tensors and the inverse of a fourth-order tensor. However, tensor operations in index notation do not translate directly into matrix operations in contracted form. For example, the double contraction of two fourth-order tensors is

$$\begin{aligned} \mathbf{C} &= \mathbf{A} : \mathbf{B} \\ C_{ijkl} &= A_{ijmn} B_{mnkl} \end{aligned} \tag{A.9}$$

Let $[a]$, $[b]$, and $[c]$ be the 6 × 6 matrix representations of the above tensors. Then, it can be shown that

$$\begin{aligned} [a]\,[b] &\neq [c] \text{ or} \\ a_{\alpha\beta} b_{\beta\gamma} &\neq c_{\alpha\gamma} \text{ (matrix multiplication)} \end{aligned} \tag{A.10}$$

The rest of this appendix presents formulas for adequate representation of tensor operations in their contracted form.

A.4 Double Contraction

In (A.9), an element like C_{1211} can be expanded as

$$\begin{aligned} C_{1211} \quad = \quad & A_{1211}B_{1111} + A_{1222}B_{2211} + A_{1233}B_{3311} \\ & +2A_{1212}B_{1211} + 2A_{1213}B_{1311} + 2A_{1223}B_{2311} \end{aligned} \tag{A.11}$$

In order to achieve the same result by matrix multiplication, multiply the last three columns of the matrix $[a]$ by 2, and then perform the multiplication

$$\begin{aligned} [c] \quad = \quad & \begin{bmatrix} A_{1111} & A_{1122} & A_{1133} & 2A_{1123} & 2A_{1113} & 2A_{1112} \\ A_{2211} & A_{2222} & A_{2233} & 2A_{2223} & 2A_{2213} & 2A_{2212} \\ A_{3311} & A_{3322} & A_{3333} & 2A_{3323} & 2A_{3313} & 2A_{3312} \\ A_{2311} & A_{2322} & A_{2333} & 2A_{2323} & 2A_{2313} & 2A_{2312} \\ A_{1311} & A_{1322} & A_{1333} & 2A_{1323} & 2A_{1313} & 2A_{1312} \\ A_{1211} & A_{1222} & A_{1233} & 2A_{1223} & 2A_{1213} & 2A_{1212} \end{bmatrix} \\ & \begin{bmatrix} B_{1111} & B_{1122} & B_{1133} & B_{1123} & B_{1113} & B_{1112} \\ B_{2211} & B_{2222} & B_{2233} & B_{2223} & B_{2213} & B_{2212} \\ B_{3311} & B_{3322} & B_{3333} & B_{3323} & B_{3313} & B_{3312} \\ B_{2311} & B_{2322} & B_{2333} & B_{2323} & B_{2313} & B_{2312} \\ B_{1311} & B_{1322} & B_{1333} & B_{1323} & B_{1313} & B_{1312} \\ B_{1211} & B_{1222} & B_{1233} & B_{1223} & B_{1213} & B_{1212} \end{bmatrix} \end{aligned} \tag{A.12}$$

This transformation can be produced by using the Reuter matrix [R]

$$[R] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix} \tag{A.13}$$

Substituting in (A.12) we have

$$[c] = [a]\,[R]\,[b] \tag{A.14}$$

A.5 Tensor Inversion

First, it is convenient to define the fourth-order identity tensor I_{ijkl} as a tensor that multiplied innerly by another fourth-order tensor yields this same tensor, or

$$I_{ijmn}A_{mnkl} = A_{ijkl} \tag{A.15}$$

If A_{ijkl} has a minor symmetry, the following tensor achieves (A.15)

$$I_{ijkl} = \frac{1}{2}\left(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}\right) \tag{A.16}$$

where δ_{ij} is the Kronecker delta, defined as

$$\begin{aligned} \delta_{ij} &= 1 \quad \text{if } i = j \\ \delta_{ij} &= 0 \quad \text{if } i \neq j \end{aligned} \tag{A.17}$$

In Voigt contracted notation, the fourth-order identity tensor is denoted as $[i]$, which is equal to the inverse of the Reuter matrix

$$[i] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/2 \end{bmatrix} = [R]^{-1} \tag{A.18}$$

Now, the inverse of a tensor is a tensor that multiplied by the original tensor yields the identity tensor, as follows:

$$A_{ijmn} A^{-1}_{mnkl} = I_{ijkl} \tag{A.19}$$

Let us introduce the following notation:

$[a]^{-1}$ = inverse of the contracted form of A_{ijkl}

$\left[a^{-1}\right]$ = contracted form of the inverse of A_{ijkl}

If A_{ijkl} has a minor symmetry, the components of $a^{-1}_{\alpha\beta}$ are:

i. Multiply the last three columns of $[a]$ by 2 by using the matrix $[R]$.

ii. Invert the obtained matrix.

iii. Multiply the matrix by $[i]$.

In orther words, the matrix $\left[a^{-1}\right]$ is computed as

$$\left[a^{-1}\right] = \left[[a]\,[R]\right]^{-1}[i] = [i]\,[a]^{-1}[i] \tag{A.20}$$

A.6 Tensor Differentiation

A.6.1 Derivative of a Tensor with Respect to Itself

Any symmetric second-order tensor Φ_{ij} satisfies the following:

$$d\Phi_{ij} = d\Phi_{ji} \tag{A.21}$$

Therefore, differentiating a second-order symmetric tensor with respect to itself is accomplished as follows

$$\frac{\partial \Phi_{ij}}{\partial \Phi_{kl}} = J_{ijkl} \tag{A.22}$$

where J_{ijkl} is a fourth-order tensor defined as

$$\begin{array}{ll} J_{ijkl} = 1 & \text{if } i = k, \text{ and } j = l \\ J_{ijkl} = 1 & \text{if } i = l, \text{ and } j = k \\ J_{ijkl} = 0 & \quad\text{otherwise} \end{array} \tag{A.23}$$

In contracted notation, the tensor J_{ijkl} is represented by

$$[j] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.24}$$

A.6.2 Derivative of the Inverse of a Tensor with Respect to the Tensor

A second-order tensor contracted with its inverse yields the second-order identity tensor, or Kronecker delta

$$A_{ij} A_{jk}^{-1} = \delta_{ik} \tag{A.25}$$

Differentiating (A.25) with respect to A_{mn} and rearranging terms yields

$$A_{ij} \frac{\partial A_{jk}^{-1}}{\partial A_{mn}} = -\frac{\partial A_{ij}}{\partial A_{mn}} A_{jk}^{-1} \tag{A.26}$$

Pre-multiplying both sides by $A_{li}^{-1}$and rearranging yields

$$\frac{\partial A_{ij}^{-1}}{\partial A_{mn}} = -A_{ik}^{-1} \frac{\partial A_{kl}}{\partial A_{mn}} A_{lj}^{-1} \tag{A.27}$$

Finally, using (A.22) yields

$$\frac{\partial A_{ij}^{-1}}{\partial A_{mn}} = -A_{ik}^{-1} J_{klmn} A_{lj}^{-1} \tag{A.28}$$

Appendix B

Second-Order Diagonal Damage Models

Explicit expressions associated to second-order diagonal damage models are presented here for completeness.

B.1 Effective and Damaged Spaces

A *second-order damage tensor* can be represented as a diagonal tensor (see (8.61))

$$D_{ij} = d_i\,\delta_{ij} \quad ; \quad \text{no sum on } i \tag{B.1}$$

in a coordinate system coinciding with the principal directions of $\mathbf{D}$, which may coincide with the fiber, transverse, and thickness directions, and d_i are the eigenvalues of the damage tensor, which represent the damage ratio along these directions. The dual variable of the damage tensor is the *integrity tensor*, $\mathbf{\Omega} = \sqrt{\mathbf{I} - \mathbf{D}}$, which represents the undamaged ratio.

The *second-order damage tensor* $\mathbf{D}$ and the *integrity tensor* $\mathbf{\Omega}$ are diagonal and have the following explicit forms

$$D_{ij} = \begin{bmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{bmatrix} \tag{B.2}$$

$$\Omega_{ij} = \begin{bmatrix} \sqrt{1-d_1} & 0 & 0 \\ 0 & \sqrt{1-d_2} & 0 \\ 0 & 0 & \sqrt{1-d_3} \end{bmatrix} = \begin{bmatrix} \Omega_1 & 0 & 0 \\ 0 & \Omega_2 & 0 \\ 0 & 0 & \Omega_3 \end{bmatrix} \tag{B.3}$$

A symmetric fourth-order tensor, $\mathbf{M}$, called the *damage effect tensor*, is defined (see (8.63)) as

$$M_{ijkl} = \frac{1}{2}\left(\Omega_{ik}\Omega_{jl} + \Omega_{il}\Omega_{jk}\right) \tag{B.4}$$

The *damage effect tensor* in contracted form multiplied by the Reuter matrix takes the form of a 6 × 6 array as follows

$$\mathbf{M} = \mathbf{M}_{\alpha\beta} = \begin{bmatrix} \Omega_1^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \Omega_2^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \Omega_3^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \Omega_2\Omega_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & \Omega_1\Omega_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & \Omega_1\Omega_2 \end{bmatrix} \tag{B.5}$$

The damaged stiffness tensor $\mathbf{C}$ multiplied by the Reuter matrix can be written in explicit contracted notation for an orthotropic material by a 6 × 6 array as a function of the undamaged stiffness tensor $\overline{\mathbf{C}}$ as follows

$$\mathbf{C}_{\alpha\beta} = \begin{bmatrix} \overline{C}_{11}\Omega_1^4 & \overline{C}_{12}\Omega_1^2\Omega_2^2 & \overline{C}_{13}\Omega_1^2\Omega_3^2 & 0 & 0 & 0 \\ \overline{C}_{12}\Omega_1^2\Omega_2^2 & \overline{C}_{22}\Omega_2^4 & \overline{C}_{23}\Omega_2^2\Omega_3^2 & 0 & 0 & 0 \\ \overline{C}_{13}\Omega_1^2\Omega_3^2 & \overline{C}_{23}\Omega_2^2\Omega_3^2 & \overline{C}_{33}\Omega_3^4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2\overline{C}_{44}\Omega_2^2\Omega_3^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2\overline{C}_{55}\Omega_1^2\Omega_3^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2\overline{C}_{66}\Omega_1^2\Omega_2^2 \end{bmatrix} \tag{B.6}$$

where $\overline{C}_{44} = \overline{G}_{23}$, $\overline{C}_{55} = \overline{G}_{13}$, and $\overline{C}_{66} = \overline{G}_{12}$. The Voigt contracted notation for fourth-order elasticity tensors is used here: $C_{\alpha\beta}$ replaces C_{ijkl} where α, β take the values 1, 2, 3, 4, 5, 6, corresponding to the index pairs 11, 22, 33, 23, 13 and 12, respectively.

The relations between the effective and actual stress components assume the following expressions

$$\begin{aligned} \overline{\sigma}_1 &= \sigma_1\,\Omega_1^{-2}; \qquad & \overline{\sigma}_4 &= \sigma_4\,\Omega_2^{-1}\Omega_3^{-1}; \\ \overline{\sigma}_2 &= \sigma_2\,\Omega_2^{-2}; & \overline{\sigma}_5 &= \sigma_5\,\Omega_1^{-1}\Omega_3^{-1}; \\ \overline{\sigma}_3 &= \sigma_3\,\Omega_3^{-2}; & \overline{\sigma}_6 &= \sigma_6\,\Omega_1^{-1}\Omega_2^{-1}; \end{aligned} \tag{B.7}$$

and the strain components

$$\begin{aligned} \overline{\varepsilon}_1 &= \varepsilon_1\,\Omega_1^2; \qquad & \overline{\varepsilon}_4 &= \varepsilon_4\,\Omega_2\Omega_3; \\ \overline{\varepsilon}_2 &= \varepsilon_2\,\Omega_2^2; & \overline{\varepsilon}_5 &= \varepsilon_5\,\Omega_1\Omega_3; \\ \overline{\varepsilon}_3 &= \varepsilon_3\,\Omega_3^2; & \overline{\varepsilon}_6 &= \varepsilon_6\,\Omega_1\Omega_2; \end{aligned} \tag{B.8}$$

where the over-line indicates an effective property.

B.2 Thermodynamic Force Y

By satisfying the Clausius-Duhem inequality, thus assuring non-negative dissipation, the following thermodynamic forces (see (8.128)) are defined

$$Y_{ij} = -\frac{\partial\psi}{\partial D_{ij}} = -\frac{1}{2}\left(\varepsilon_{kl}-\varepsilon_{kl}^p\right)\frac{\partial C_{klpq}}{\partial D_{ij}}\left(\varepsilon_{pq}-\varepsilon_{pq}^p\right) = -\frac{1}{2}\varepsilon_{kl}^e\frac{\partial C_{klpq}}{\partial D_{ij}}\varepsilon_{pq}^e \tag{B.9}$$

The second-order tensor of the conjugate thermodynamic forces associated to the damage variables takes the following form

$$\mathbf{Y} = Y_{ij} = \begin{bmatrix} Y_{11} & 0 & 0 \\ 0 & Y_{22} & 0 \\ 0 & 0 & Y_{33} \end{bmatrix} \tag{B.10}$$

or in Voigt contracted notation as

$$\mathbf{Y} = Y_\alpha = \{Y_{11}, Y_{22}, Y_{33}, 0, 0, 0\}^T \tag{B.11}$$

Using (B.9), the explicit expressions for the thermodynamic forces written in terms of effective strain are found as

$$\begin{aligned} Y_{11} &= \frac{1}{\Omega_1^2}\left(\overline{C}_{11}\,\overline{\varepsilon}_1^{e2} + \overline{C}_{12}\,\overline{\varepsilon}_2^e\,\overline{\varepsilon}_1^e + \overline{C}_{13}\,\overline{\varepsilon}_3^e\,\overline{\varepsilon}_1^e + 2\overline{C}_{55}\,{\overline{\varepsilon}_5^e}^2 + 2\overline{C}_{66}\,{\overline{\varepsilon}_6^e}^2\right) \\ Y_{22} &= \frac{1}{\Omega_2^2}\left(\overline{C}_{22}\,\overline{\varepsilon}_2^{e2} + \overline{C}_{12}\,\overline{\varepsilon}_2^e\,\overline{\varepsilon}_1^e + \overline{C}_{23}\,\overline{\varepsilon}_3^e\,\overline{\varepsilon}_2^e + 2\overline{C}_{44}\,{\overline{\varepsilon}_4^e}^2 + 2\overline{C}_{66}\,{\overline{\varepsilon}_6^e}^2\right) \\ Y_{33} &= \frac{1}{\Omega_3^2}\left(\overline{C}_{33}\,\overline{\varepsilon}_3^{e2} + \overline{C}_{13}\,\overline{\varepsilon}_3^e\,\overline{\varepsilon}_1^e + \overline{C}_{23}\,\overline{\varepsilon}_3^e\,\overline{\varepsilon}_2^e + 2\overline{C}_{44}\,{\overline{\varepsilon}_4^e}^2 + 2\overline{C}_{55}\,{\overline{\varepsilon}_5^e}^2\right) \end{aligned} \tag{B.12}$$

The thermodynamic forces written in terms of actual stress are

$$\begin{aligned} Y_{11} &= \frac{1}{\Omega_1^2}\left(\frac{\overline{S}_{11}}{\Omega_1^4}{\sigma_1}^2 + \frac{\overline{S}_{12}}{\Omega_1^2\Omega_2^2}\sigma_2\sigma_1 + \frac{\overline{S}_{13}}{\Omega_1^2\Omega_3^2}\sigma_3\sigma_1 + \frac{2\overline{S}_{55}}{\Omega_1^2\Omega_3^2}{\sigma_5}^2 + \frac{2\overline{S}_{66}}{\Omega_1^2\Omega_2^2}{\sigma_6}^2\right) \\ Y_{22} &= \frac{1}{\Omega_2^2}\left(\frac{\overline{S}_{22}}{\Omega_2^4}{\sigma_2}^2 + \frac{\overline{S}_{12}}{\Omega_2^2\Omega_1^2}\sigma_2\sigma_1 + \frac{\overline{S}_{23}}{\Omega_2^2\Omega_3^2}\sigma_3\sigma_2 + \frac{2\overline{S}_{44}}{\Omega_2^2\Omega_3^2}{\sigma_4}^2 + \frac{2\overline{S}_{66}}{\Omega_2^2\Omega_1^2}{\sigma_6}^2\right) \\ Y_{33} &= \frac{1}{\Omega_3^2}\left(\frac{\overline{S}_{33}}{\Omega_3^4}{\sigma_3}^2 + \frac{\overline{S}_{13}}{\Omega_3^2\Omega_1^2}\sigma_3\sigma_1 + \frac{\overline{S}_{23}}{\Omega_3^2\Omega_2^2}\sigma_3\sigma_2 + \frac{2\overline{S}_{44}}{\Omega_3^2\Omega_2^2}{\sigma_4}^2 + \frac{2\overline{S}_{55}}{\Omega_3^2\Omega_1^2}{\sigma_5}^2\right) \end{aligned} \tag{B.13}$$

The derivative of the thermodynamic forces with respect to the damage $(\partial\mathbf{Y}/\partial\mathbf{D})$ is given by

$$\frac{\partial\mathbf{Y}}{\partial\mathbf{D}} = \begin{bmatrix} \dfrac{Y_{11}}{\Omega_1^4} & 0 & 0 & 0 & 0 & 0 \\ 0 & \dfrac{Y_{22}}{\Omega_2^4} & 0 & 0 & 0 & 0 \\ 0 & 0 & \dfrac{Y_{33}}{\Omega_3^4} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{B.14}$$

The derivative of the thermodynamic forces with respect to the actual strain is given by

$$\frac{\partial \mathbf{Y}}{\partial \boldsymbol{\varepsilon}^e} = \begin{bmatrix} -\frac{P_{11}}{\Omega_1^2} & -\frac{\overline{C}_{12}\,\overline{\varepsilon}_1^e}{\Omega_1^2} & -\frac{\overline{C}_{13}\,\overline{\varepsilon}_1^e}{\Omega_1^2} & 0 & -2\,\frac{\overline{C}_{55}\,\overline{\varepsilon}_5^e}{\Omega_1^2} & -2\,\frac{\overline{C}_{66}\,\overline{\varepsilon}_6^e}{\Omega_1^2} \\ -\frac{\overline{C}_{12}\,\overline{\varepsilon}_2^e}{\Omega_2^2} & -\frac{P_{22}}{\Omega_2^2} & -\frac{\overline{C}_{23}\,\overline{\varepsilon}_2^e}{\Omega_2^2} & -2\,\frac{\overline{C}_{44}\,\overline{\varepsilon}_4^e}{\Omega_2^2} & 0 & -2\,\frac{\overline{C}_{66}\,\overline{\varepsilon}_6^e}{\Omega_2^2} \\ -\frac{\overline{C}_{13}\,\overline{\varepsilon}_3^e}{\Omega_3^2} & -\frac{\overline{C}_{23}\,\overline{\varepsilon}_3^e}{\Omega_3^2} & -\frac{P_{33}}{\Omega_3^2} & -2\,\frac{\overline{C}_{44}\,\overline{\varepsilon}_4^e}{\Omega_3^2} & -2\,\frac{\overline{C}_{55}\,\overline{\varepsilon}_5^e}{\Omega_3^2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{B.15}$$

where

$$\begin{aligned} P_{11} = &\; 2\,\overline{C}_{11}\,\overline{\varepsilon}_1^e + \overline{C}_{12}\,\overline{\varepsilon}_2^e + \overline{C}_{13}\,\overline{\varepsilon}_3^e \\ P_{22} = &\; \overline{C}_{12}\,\overline{\varepsilon}_1^e + 2\,\overline{C}_{22}\,\overline{\varepsilon}_2^e + \overline{C}_{23}\,\overline{\varepsilon}_3^e \\ P_{33} = &\; \overline{C}_{13}\,\overline{\varepsilon}_1^e + \overline{C}_{23}\,\overline{\varepsilon}_2^e + 2\,\overline{C}_{33}\,\overline{\varepsilon}_3^e \end{aligned} \tag{B.16}$$

The derivative of the thermodynamic forces with respect to the actual unrecoverable strain is given by

$$\frac{\partial \mathbf{Y}}{\partial \boldsymbol{\varepsilon}^p} = -\frac{\partial \mathbf{Y}}{\partial \boldsymbol{\varepsilon}^e} \tag{B.17}$$

The derivative of the actual stress with respect to damage is given by

$$\frac{\partial \boldsymbol{\sigma}}{\partial \mathbf{D}} = \begin{bmatrix} P'_{11} & 0 & 0 & 0 & 0 & 0 \\ 0 & P'_{22} & 0 & 0 & 0 & 0 \\ 0 & 0 & P'_{33} & 0 & 0 & 0 \\ 0 & -\frac{1}{2}\frac{\Omega_3\overline{C}_{44}\,\overline{\varepsilon}_4^e}{\Omega_2} & -\frac{1}{2}\frac{\Omega_2\overline{C}_{44}\,\overline{\varepsilon}_4^e}{\Omega_3} & 0 & 0 & 0 \\ -\frac{1}{2}\frac{\Omega_3\overline{C}_{55}\,\overline{\varepsilon}_5^e}{\Omega_1} & 0 & -\frac{1}{2}\frac{\Omega_1\overline{C}_{55}\,\overline{\varepsilon}_5^e}{\Omega_3} & 0 & 0 & 0 \\ -\frac{1}{2}\frac{\Omega_2\overline{C}_{66}\,\overline{\varepsilon}_6^e}{\Omega_1} & -\frac{1}{2}\frac{\Omega_1\overline{C}_{66}\,\overline{\varepsilon}_6^e}{\Omega_2} & 0 & 0 & 0 & 0 \end{bmatrix} \tag{B.18}$$

where

$$\begin{aligned} P'_{11} = &\; -\overline{C}_{11}\,\overline{\varepsilon}_1^e - \overline{C}_{12}\,\overline{\varepsilon}_2^e - \overline{C}_{13}\,\overline{\varepsilon}_3^e \\ P'_{22} = &\; -\overline{C}_{12}\,\overline{\varepsilon}_1^e - \overline{C}_{22}\,\overline{\varepsilon}_2^e - \overline{C}_{23}\,\overline{\varepsilon}_3^e \\ P'_{33} = &\; -\overline{C}_{13}\,\overline{\varepsilon}_1^e - \overline{C}_{23}\,\overline{\varepsilon}_2^e - \overline{C}_{33}\,\overline{\varepsilon}_3^e \end{aligned} \tag{B.19}$$

B.3 Damage Surface

An anisotropic damage criterion expressed in tensorial form, introducing two fourth-order tensors, **B** and **J**, defines a multiaxial limit surface in the thermodynamic force space, **Y**, that bounds the damage domain. The damage evolution is defined by a damage potential associate to the damage surface and by an isotropic hardening function. The proposed damage surface g^d is given by

$$g^d = \left(\hat{Y}_{ij}^N J_{ijhk} \hat{Y}_{hk}^N\right)^{1/2} + \left(Y_{ij}^S B_{ijhk} Y_{hk}^S\right)^{1/2} - (\gamma(\delta) + \gamma_0) \tag{B.20}$$

where γ_0 is the initial damage threshold value and $\gamma(\delta)$ defines the hardening.

The derivative of the damage surface with respect to thermodynamic forces is given by

$$\frac{\partial g^d}{\partial \mathbf{Y}} = \begin{bmatrix} \dfrac{J_{11}\, Y_{11}^N}{\Phi^N} + \dfrac{B_{11}\, Y_{11}^S}{\Phi^S} \\ \dfrac{J_{22}\, Y_{22}^N}{\Phi^N} + \dfrac{B_{22}\, Y_{22}^S}{\Phi^S} \\ \dfrac{J_{33}\, Y_{33}^N}{\Phi^N} + \dfrac{B_{33}\, Y_{33}^S}{\Phi^S} \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{B.21}$$

where

$$\begin{aligned} \Phi^N &= \sqrt{J_{11}\,(Y_{11}^N)^2 + J_{22}\,(Y_{22}^N)^2 + J_{33}\,(Y_{33}^N)^2} \\ \Phi^S &= \sqrt{B_{11}\,(Y_{11}^S)^2 + B_{22}\,(Y_{22}^S)^2 + B_{33}\,(Y_{33}^S)^2} \end{aligned} \tag{B.22}$$

The derivative of the damage surface with respect to damage hardening is

$$\frac{\partial g^d}{\partial \gamma} = -1 \tag{B.23}$$

B.4 Unrecoverable-Strain Surface

The unrecoverable-strain (yield) surface g^p is a function of the thermodynamic forces in the effective configuration $(\overline{\sigma}, R)$. Therefore, the unrecoverable-strain surface is

$$g^p = \sqrt{f_{ij}\overline{\sigma}_i\overline{\sigma}_j + f_i\overline{\sigma}_i} - (R\,(p) + R_0) \tag{B.24}$$

where $(i = 1, 2...6)$, R_0 is the initial unrecoverable-strain threshold, and R is the hardening function.

The derivative of the unrecoverable-strain surface with respect to effective stress is given by

$$\frac{\partial \mathbf{g^p}}{\partial \overline{\sigma}} = \begin{bmatrix} \dfrac{1}{2}\,\dfrac{f_1 + 2\,f_{11}\,\overline{\sigma}_1 + 2\,f_{12}\,\overline{\sigma}_2 + 2\,f_{13}\,\overline{\sigma}_3}{\Phi^p} \\ \dfrac{1}{2}\,\dfrac{f_2 + 2\,f_{22}\,\overline{\sigma}_2 + 2\,f_{12}\,\overline{\sigma}_1 + 2\,f_{23}\,\overline{\sigma}_3}{\Phi^p} \\ \dfrac{1}{2}\,\dfrac{f_3 + 2\,f_{33}\,\overline{\sigma}_3 + 2\,f_{13}\,\overline{\sigma}_1 + 2\,f_{23}\,\overline{\sigma}_2}{\Phi^p} \\ \dfrac{f_4\,\overline{\sigma}_4}{\Phi^p} \\ \dfrac{f_5\,\overline{\sigma}_5}{\Phi^p} \\ \dfrac{f_6\,\overline{\sigma}_6}{\Phi^p} \end{bmatrix} \tag{B.25}$$

where

$$\begin{aligned}\Phi^p \quad &= (f_1\,\overline{\sigma}_1 + f_2\,\overline{\sigma}_2 + f_3\,\overline{\sigma}_3 + \\ &+ f_{11}\,\overline{\sigma}_1{}^2 + f_{22}\,\overline{\sigma}_2{}^2 + f_{33}\,\overline{\sigma}_3{}^2 + \\ &+ 2\,f_{12}\,\overline{\sigma}_1\,\overline{\sigma}_2 + 2\,f_{13}\,\overline{\sigma}_1\,\overline{\sigma}_3 + 2\,f_{23}\,\overline{\sigma}_2\,\overline{\sigma}_3 + \\ &+ f_6\,\overline{\sigma}_6{}^2 + f_5\,\overline{\sigma}_5{}^2 + f_4\,\overline{\sigma}_4{}^2\big)^{1/2}\end{aligned} \tag{B.26}$$

The derivative of the yield surface with respect to unrecoverable-strain hardening is

$$\frac{\partial g^p}{\partial R} = -1 \tag{B.27}$$

Appendix C

Software Used

Only four software applications are used throughout this textbook. *Abaqus* is by far the most used. BMI3 is used only in Chapter 4. *MATLAB* is used for symbolic as well as numerical computations. Finally, *Intel Fortran* must be available to compile and link *Abaqus* with user programmed material subroutines, but its usage is transparent to the user because it is called by a batch file requiring no user intervention. Of course, some knowledge of *Fortran* is required to program new user material subroutines, but programming is made easier by several example subroutines, which are provided and used in the examples.

The aim of this section is to present an introduction to the software used in this textbook, namely *Abaqus* and BMI3, as well as how to use *Intel Fortran* to compile and link user subroutines with *Abaqus*. It is assumed that the reader can use *MATLAB* without help besides that provided by the self-explanatory *MATLAB* code included with the examples, either printed in this textbook or downloadable from the Web site [5].

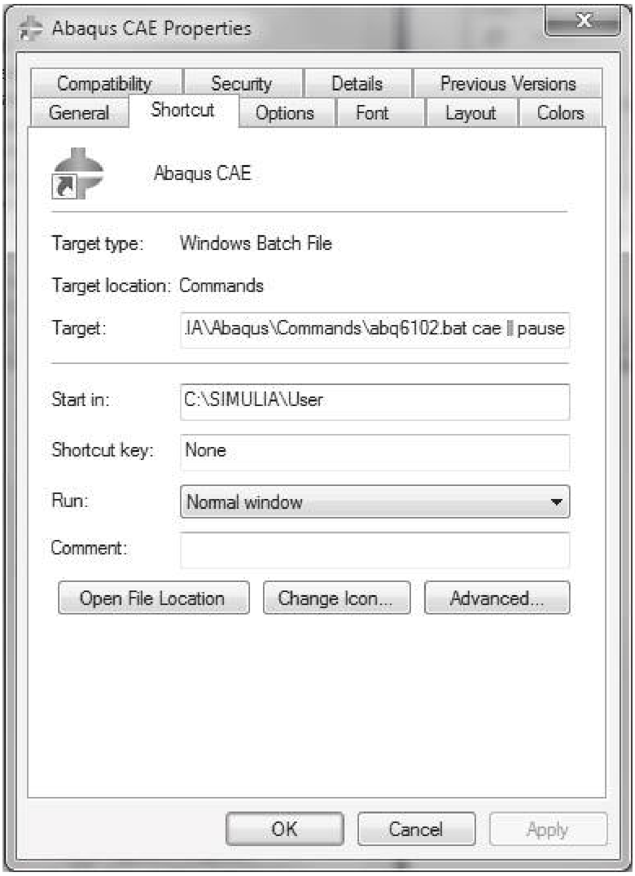
Operation of the software is illustrated for a *Windows 7* platform but operation in a *Linux* environment is similar. For the sake of space, this section is very brief. The vendors of these applications have a wealth of information, training sessions, user groups, and so on, that the reader can use to get familiar with the software interface. One such source of information is the Web site for this textbook at `http://barbero.cadec-online.com/feacm-abaqus` [5]. Another source of information is the textbook's forum [194].

C.1 *Abaqus*

Abaqus is a commercial finite element analysis (FEA) application. It has a friendly graphical user interface (GUI) called *CAE* and an extensive documentation system. Once started, the user should have no difficulty navigating menus and so on.

All the mouse clicks in the GUI generate CAE script, which are saved in several files, but the script is very difficult to understand without significant time investment in learning CAE scripting [195].

The examples in this textbook were produced using *Abaqus* version 6.10. They



Target type: Windows Batch File

Target location: Commands

Target: IA\Abaqus\Commands\abq6102.bat cae || pause

Start in: C:\SIMULIA\User

Shortcut key: None

Run: Normal window

Comment:

Fig. C.1: Edit the shortcut to change the default Work directory.

work identically in *Windows 7* and *Linux* platforms. On *Windows 7*, *Abaqus* is accessible from the START menu, through `Start, All Programs, Abaqus 6.10, Abaqus CAE`.

In this textbook, it is assumed that the user has created a folder `c:\Simulia\User`, where the model files will reside.

If the GUI is started using the shortcut `Start, All Programs, Abaqus 6.10`, the temporary files reside in the default `Work directory`, which was set during installation of the *Abaqus* software, typically `c:\temp`. The default work directory can be changed by editing the `Start in:` field in the shortcut, as shown in Figure C.1. Otherwise, the Work Directory for the current session can be changed from within *Abaqus*/CAE by doing the following:

```
Menu:   File, Set Work Directory
```

The default directory is used by *Abaqus* as a default location to store temporary files, including:

.dat The data file contains a printout of requested results.

.fil The result file contains results such as mode shapes and so on (see Example 4.3, p. 188). This file is unformatted.

.inp The input file contains the input to the *Abaqus* processor.

.jnl The journal file contains changes to the .mdb since the last `Save`.

.log The log file contains a brief log of the last execution of the *Abaqus* processor.

.odb The output database file contains all the results from an execution of the *Abaqus* processor.

.rec The recovery file contains a record of the operations performed by the user on the GUI since the last `Save`.

.rpy The replay file contains a replay of the whole procedure that you used in the GUI to generate the model.

In this textbook, we prefer to store and overwrite temporary files in the default `Work directory` to maintain clean user folders, but sometimes we will purposely store temporary files in a user folder. For example, if we want to read some output from the `.dat` file, we will store it in a user folder.

This section attempts to describe the installation and "systems" aspects of *Abaqus* software. On the other hand, a description of the *Abaqus*/CAE GUI is integrated within the examples in the textbook, starting with Chapter 3. Therefore, it is highly recommended that the user installs *Abaqus*, then proceeds to study Chapter 3 and personally "do" all the examples in Chapter 3. By the end of Chapter 3, you should be quite skilled in *Abaqus*/CAE, at least as far is concerned with composites modeling. Even higher proficiency will be achieved by studying and practicing the examples in the entire textbook. It has been my experience that students successfully teach themselves *Abaqus*/CAE by working the examples in this textbook, along with the help system and the documentation included with *Abaqus*. Another source of help is the video recordings illustrating the execution of the examples by using the *Abaqus*/CAE GUI [5].

C.1.1 *Abaqus* Programmable Features

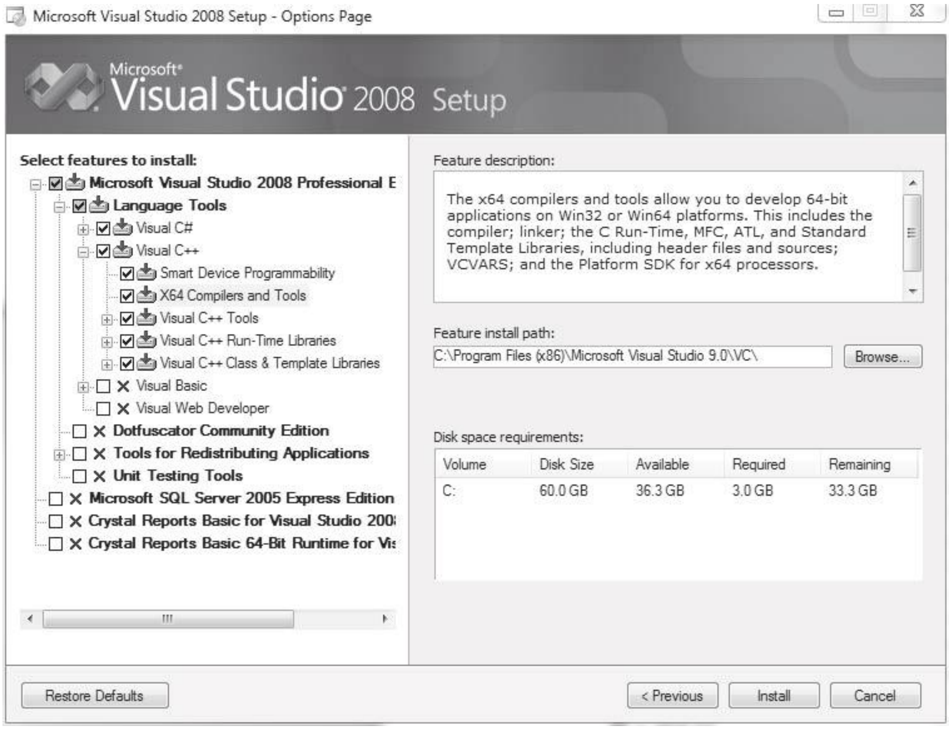
To prepare *Abaqus* for running UMAT subroutines and other *Abaqus* programmable features, the following software must be installed in this order:

i. *Microsoft Visual Studio© (VS) 2008.*

ii. *Intel Fortran© 11.1.*

First, install *Microsoft Visual Studio 2008* (VS2008) and update it to service pack 1 (SP1) with the update available on the Web.

Next, install *Intel Fortran 11.1.* Fortran must be installed after VS so that VS is ready to integrate Fortran into it. During installation, install the X64 Compiler and Tools, which are not installed by default, by check marking the box *X64 Compilers and Tools* as shown in Figure C.2.



Microsoft Visual Studio 2008 Setup - Options Page
Microsoft®
Visual Studio® 2008 Setup
Browse...
Restore Defaults
< Previous
Install
Cancel

Select features to install:

- ☑ Microsoft Visual Studio 2008 Professional E
  - ☑ Language Tools
    - ☑ Visual C#
    - ☑ Visual C++
      - ☑ Smart Device Programmability
      - ☑ X64 Compilers and Tools
      - ☑ Visual C++ Tools
      - ☑ Visual C++ Run-Time Libraries
      - ☑ Visual C++ Class & Template Libraries
    - ☐ ✕ Visual Basic
    - ☐ ✕ Visual Web Developer
  - ☐ ✕ Dotfuscator Community Edition
  - ☐ ✕ Tools for Redistributing Applications
  - ☐ ✕ Unit Testing Tools
- ☐ ✕ Microsoft SQL Server 2005 Express Edition
- ☐ ✕ Crystal Reports Basic for Visual Studio 200
- ☐ ✕ Crystal Reports Basic 64-Bit Runtime for Vi

Feature description:

The x64 compilers and tools allow you to develop 64-bit applications on Win32 or Win64 platforms. This includes the compiler; linker; the C Run-Time, MFC, ATL, and Standard Template Libraries, including header files and sources; VCVARS; and the Platform SDK for x64 processors.

Feature install path:

C:\Program Files (x86)\Microsoft Visual Studio 9.0\VC\

Disk space requirements:

Volume	Disk Size	Available	Required	Remaining
C:	60.0 GB	36.3 GB	3.0 GB	33.3 GB

Fig. C.2: Visual Studio 2008. Setup page.

Next, locate the batch file:
`c:\SIMULIA\Abaqus\Commands\abq6102.bat`
Make a backup copy. Then, modify it as follows (also available in [5]).

```
rem Windows 64 version of .bat file to execute Abaqus
rem with User Programmable Features and Intel MKL Math libraries

SET PATH=C:\Program Files (x86)\Microsoft Visual Studio 9.0\VC\Bin;%PATH%;
SET PATH=C:\Program Files (x86)\Intel\Compiler\11.1\048\bin\Intel64;%PATH%;
SET PATH=C:\Program Files (x86)\Common Files\Microsoft Shared\VSA\9.0\VsaEnv;
%PATH%;

SET LIB=C:\Program Files (x86)\Microsoft Visual Studio 9.0\VC\Lib;%LIB%
SET LIB=C:\Program Files (x86)\Intel\Compiler\11.1\048\mkl\em64t\lib;%LIB%

call "C:\Program Files (x86)\Intel\Compiler\11.1\048\bin\intel64
\ifortvars_intel64"

rem keep next lines exactly as in your original installation
@echo off
call "C:\SIMULIA\Abaqus\6.10-1\exec\abq6101.exe" %*
```

Note that file names and locations may be slightly different in your installation, especially if you are using a 32-Bit system. If you modify the `abq6102.bat`, you must use the exact location of the files in your system. Failing to do so will cause *Abaqus* to prompt an error message while trying to compile/link UMAT subroutines.

Next, locate the environment file:
`C:\SIMULIA\Abaqus\6.10-2\site\abaqus_v6.env`
Make a backup copy. Then, modify the `compile_fortran` directive to include three additional switches. The first two, `/DABQ_WIN86_64` and `/Qmkl:sequential`, allow use of the Intel math library `MKL`. The third, `free`, allows *Abaqus* to understand free formatted fortran code in the user subroutines even though *Abaqus* requires `.for` extension for any user supplied Fortran code. For a Windows 64 bit system, the resulting section of the environment file should look something like this:

```
compile_fortran=['ifort','/c','/DABQ_WIN86_64','/Qmkl:sequential','/free',
      '/recursive','/Qauto-scalar','/QxW','/nologo','/Od','/include:%I']
```

Using Visual Studio 2008, you should debug your Fortran code before linking it to *Abaqus*. A simple program such as `umat_tester.f90`, which is available in [5], can be used to test your user material subroutine before linking it to *Abaqus*. You can modify the `umat_tester.f90` to suit your needs.

Fortran code can be written in *free* format as long as the file name given to *Abaqus* has an extension `.for` or it is compiled into object code `.obj` using the `abaqus make` utility. By default, VS2008 interprets files having extension `.for` as containing 72 column, fixed format, Fortran 77 code. Since F77 is very limited, free format, Fortran 90 is preferred. VS2008 is be able to work with a `.for` file even if it contains free formatted code, provided you add the compiler directive `!DEC$ FREEFORM` as the first line to the code. Otherwise, the VS2008 default behavior can be changed inside VS2008. In addition to that, the "`free`" switch should be added to the `compile_fortran` directive in the environment file so that *Abaqus* can deal with free formatted code.

`UMAT`, `VUMAT`, `UGENS`, and `UEL` examples in Abaqus documentation are written in F77. A free converter, written in F90, is available in [196].

An Errata for the textbook is maintained in [5]. Known installation issues will be reported there.

C.2 BMI3

Native BMI3 code accepts an input file in *Abaqus* `.inp` format, as long as the input file is filtered by the program `inp2bmi3.exe`. Therefore, one can create the model in *Abaqus*/CAE, write the `.inp` file from the Job Manager, filter it with `inp2bmi3.exe`, and execute it with `bmi3.exe`. The executables are available in [5]. The input/output file sequence is described in Table C.1. The sequence of execution is:

– Create a model in *Abaqus*/CAE and write the `Job-1.inp` file

Table C.1: Input/Output files to execute BMI3.

Program	Read	Write	Comments
inp2bmi3.exe	Job-1.inp	BMI3.inp	filtered model
		BMI3.dat	material properties
		ABQ.inp	filtered Abaqus
BMI3.exe	BMI3.inp	BMI3.out	results
	BMI3.dat	MODES.out	mode shapes

– Execute `inp2bmi3.exe` to get the filtered files. If needed, modify the *mode, node, component* in the `BMI3.dat` file

– Execute `bmi3.exe`. Record the results from the display or from `BMI3.out`

Not all of the *Abaqus* `.inp` commands are understood by the filter `inp2bmi3.exe`. Most of the limitations could be removed by modifying the filter, for which the source code `inp2bmi3.f90` is available in [5]. The current limitations are:

– The mesh must refer to S8R or S8R5 elements (see Example 4.2, p. 185).

– All loads must be concentrated loads.

– The material properties must be given using Shell General Stiffness.

– Only one material can be used for the entire model.

– Constrained boundary conditions cannot be used.

– Materials Orientation (*orientation) cannot be used.

Abaqus/CAE writes an `.inp` file every time a job is submitted, or `Write Input` is executed from the `Job Manager` window. Then, run `inp2bmi3.exe` to generate `BMI3.inp`, `ABQ.inp`, and `BMI3.dat`. If `ABQ.inp` is executed using *Abaqus* in command mode (see Example 2.1, p. 46), or imported into *Abaqus*/CAE and executed, it will give the bifurcation loads $\Lambda^{(cr)}$ and the mode shapes.

The material properties and perturbation parameters are provided via `BMI3.dat`, which is automatically created by the filter. The last line in `BMI3.dat` contains *modenum*, *nodenum*, *component*. These are the *mode, node,* and *component* used as perturbation parameter δ. If all three values are zeros (the default), BMI3 picks the lowest mode and the node-component combination that yields the largest mode amplitude. The results of BMI3 are printed in `BMI3.out` and the mode shapes saved in the `MODES.out` file.

Note that the results (bifurcation loads, slopes, and curvatures) appear with negative sign. This is usual in stability analysis. Another peculiarity of the BMI3 software is that transverse deflections w (perpendicular to the plate) have opposite sign to *Abaqus* results. Since transverse deflections w are often used as perturbation parameters, the change in sign must be taken into account during interpretation of results (see Example 4.2, p. 185).

References

[1] E. J. Barbero. *Introduction to Composite Materials Design–Second Edition.* http://barbero.cadec-online.com/icmd. CRC Press, Boca Raton, FL, 2010.

[2] D. Frederick and T.-S. Chang. *Continuum Mechanics.* Scientific Publishers, Cambridge, MA, 1972.

[3] F. P. Beer, E. R. Johnston Jr., and J. T. DeWolf. *Mechanics of Materials, 3rd Edition.* McGraw-Hill, Boston, MA, 2001.

[4] J. N. Reddy. *Energy and Variational Methods in Applied Mechanics.* Wiley, 1984.

[5] E. J. Barbero. Web resource: http://barbero.cadec-online.com/feacm-abaqus.

[6] S. S. Sonti, E. J. Barbero, and T. Winegardner. Mechanical properties of pultruded E-glass–vinyl ester composites. In *50th Annual Conference, Composites Institute, Society of the Plastics Industry (February) pp. 10-C/1-7.*, 1995.

[7] Simulia. Abaqus documentation. The default location for the HTML documentation is http://HOME:2080/v6.10/ where HOME is the url of the server where the documentation is stored during installation. The default location for the PDF documentation is C:\SIMULIA\Documentation\docs\v6.10\pdf_books\index.pdf.

[8] E. J. Barbero. *Finite Element Analysis of Composite Materials–First Edition.* CRC Press, Boca Raton, FL, 2007.

[9] ANSYS Inc. Ansys mechanical apdl structural analysis guide, release 140.0, http://www1.ansys.com/customer/content/documentation/140/ans_str.pdf, 2011.

[10] Simulia. Abaqus keywords reference manual. The default location is http://HOME:2080/v6.10/books/key/default.htm where HOME is the url of the server where the documentation is stored during installation.

[11] SolidWorks. http://www.solidworks.com/sw/engineering-education-software.htm.

[12] E. J. Barbero. Computer aided design environment for composites. http://www.cadec-online.com, 2011.

[13] R. J. Roark and W. C. Young. *Roark's Formulas for Stress and Strain, 6th Edition.* McGraw-Hill, New York, NY, 1989.

[14] J. N. Reddy. *Mechanics of Laminated Composite Plates and Shells, 2nd Edition.* CRC Press, Boca Raton, FL, 2003.

[15] E. Hinton and D. R. J. Owen. *An Introduction to Finite Element Computations.* Pineridge Press, Swansea, UK, 1979.

[16] NAFEMS. Test R0031/3. Technical report, National Agency for Finite Element Methods and Standards (UK), 1995.

[17] E. J. Barbero. 3-d finite element for laminated composites with 2-d kinematic constraints. *Computers and Structures*, 45(2):263–271, 1992.

[18] E. J. Barbero. Cadec application program interface. `http://www.cadec-online.com/Help/API.aspx`.

[19] Simulia. Abaqus benchmarks manual. The default location is `http://HOME:2080/v6.10/books/bmk/default.htm` where `HOME` is the `url` of the server where the documentation is stored during installation.

[20] E. J. Barbero and J. Trovillion. Prediction and measurement of post-critical behavior of fiber-reinforced composite columns. *Composites Science and Technology*, 58:1335–1341, 1998.

[21] E. J. Barbero. Prediction of compression strength of unidirectional polymer matrix composites. *Journal of Composite Materials*, 32(5)(5):483–502, 1998.

[22] A. Puck and H. Schurmann. Failure analysis of frp laminates by means of physically based phenomenological models. *Composites Science and Technology*, 62:1633–1662, 2002.

[23] MIL17.org. The composite materials handbook, web resource, `http://www.mil17.org`.

[24] L. A. Godoy. *Theory of Stability-Analysis and Sensitivity.* Taylor and Francis, Philadelphia, PA, 2000.

[25] E. J. Barbero, L. A. Godoy, and I. G. Raftoyiannis. Finite elements for three-mode interaction in buckling analysis. *International Journal for Numerical Methods in Engineering*, 39(3):469–488, 1996.

[26] L. A. Godoy, E. J. Barbero, and I. G. Raftoyiannis. Finite elements for post-buckling analysis. i - the w-formulation. *Computers and Structures*, 56(6):1009–1017, 1995.

[27] E. J. Barbero, I. G. Raftoyiannis, and L. A. Godoy. Finite elements for post-buckling analysis. ii - application to composite plate assemblies. *Computers and Structures*, 56(6):1019–1028, 1995.

[28] I. G. Raftoyiannis, L. A. Godoy, and E. J. Barbero. Buckling mode interaction in composite plate assemblies. *Applied Mechanics Reviews*, 48(11/2):52–60, 1995.

[29] E. J. Barbero. Prediction of buckling-mode interaction in composite columns. *Mechanics of Composite Materials and Structures*, 7(3):269–284, 2000.

[30] S. Yamada and J. G. A. Croll. Buckling behavior pressure loaded cylindrical panels. *ASCE Journal of Engineering Mechanics*, 115(2):327–344, 1989.

[31] C. T. Herakovich. *Mechanics of Fibrous Composites.* Wiley, New York, 1998.

[32] J. D. Eshelby. The determination of the elastic field of an ellipsoidal inclusion and related problems. *Proceedings of the Royal Society*, A241:376–396, 1957.

[33] J. D. Eshelby. The elastic field outside an ellipsoidal inclusion. *Acta Metall.*, A252:561–569, 1959.

[34] T. Mori and K. Tanaka. The elastic field outside an ellipsoidal inclusion. *Acta Metall.*, 21:571–574, 1973.

[35] R. Hill. A self-consistent mechanics of composite materials. *J. Mech. Phys. Solids*, 13:213–222, 1965.

[36] S. Nemat-Nasser and M. Hori. *Micromechanics: Overall Properties of Heterogeneous Materials.* North-Holland, Amsterdam, 1993.

[37] R. Luciano and E. J. Barbero. Formulas for the stiffness of composites with periodic microstructure. *I. J. Solids and Structures*, 31(21):2933–2944, 1995.

[38] R. Luciano and E. J. Barbero. Analytical expressions for the relaxation moduli of linear viscoelastic composites with periodic microstructure. *ASME J. Applied Mechanics*, 62(3):786–793, 1995.

[39] E. J. Barbero and R. Luciano. Micromechanical formulas for the relaxation tensor of linear viscoelastic composites with transversely isotropic fibers. *I. J. Solids and Structures*, 32(13):1859–1872, 1995.

[40] J. Aboudi. *Mechanics of Composite Materials : A Unified Micromechanical Approach. Vol. 29 of Studies in Applied Mechanics.* Elsevier, New York, NY, 1991.

[41] Z. Hashin and S. Shtrikman. A variational approach to the elastic behavior of multiphase materials. *J. Mechanics and Physics of Solids*, 11:127–140, 1963.

[42] V. Tvergaard. Model studies of fibre breakage and debonding in a metal reinforced by short fibres. *J. Mechanics and Physics of Solids*, 41(8):1309–1326, 1993.

[43] J. L. Teply and G. J. Dvorak. Bound on overall instantaneous properties of elastic-plastic composites. *J. Mechanics and Physics of Solids*, 36(1):29–58, 1988.

[44] R. Luciano and E. Sacco. Variational methods for the homogenization of periodic media. *European J. Mech. A/Solids*, 17:599–617, 1998.

[45] G. J. Creus. *Viscoelasticity: Basic Theory and Applications to Concrete Structures.* Springer-Verlag, Berlin, 1986.

[46] Eric W. Weisstein. Gamma function. MathWorld–A Wolfram Web Resource. `http://mathworld.wolfram.com/GammaFunction.html`.

[47] G. D. Dean, B. E. Read, and P. E. Tomlins. A model for long-term creep and the effects of physical ageing in poly (butylene terephthalate). *Plastics and Rubber Processing and Applications*, 13(1):37–46, 1990.

[48] B. F. Oliveira and G. J. Creus. An analytical-numerical framework for the study of ageing in fibre reinforced polymer composites. *Composites B*, 65(3-4):443–457, 2004.

[49] K. Ogatha. *Discrete-Time Control System.* Prentice Hall, Englewood Cliffs, NJ, 1987.

[50] K. J. Hollenbeck. Invlap.m: A matlab function for numerical inversion of Laplace transforms by the de Hoog algorithm (1998), `http://www.mathworks.com/`.

[51] R. S. Lakes. *Viscoelastic Solids.* Boca Raton, FL, 1998.

[52] J. D. Ferry. *Viscoelastic Properties of Polymers. 3rd. Edition.* Wiley, New York, 1980.

[53] R. M. Christensen. *Theory of Viscoelasticity.* New York, 1972.

[54] P. Qiao, E. J. Barbero, and J. F. Davalos. On the linear viscoelasticity of thin-walled laminated composite beams. *Journal of Composite Materials*, 34(1):39–68, 2000.

[55] E. J. Barbero, F. A. Cosso, R. Roman, and T. L. Weadon. Determination of material parameters for Abaqus progressive damage analysis of E-Glass Epoxy laminates. *Composites Part B:Engineering*, 46(3):211–220, 2012.

[56] J. Lemaitre and A. Plumtree. Application of damage concepts to predict creep-fatigue failures. In *American Society of Mechanical Engineers, 78-PVP-26*, pages 10–26, 1978.

[57] I. N. Rabotnov. *Rabotnov: Selected Works - Problems of the Mechanics of a Deformable Solid Body.* Moscow Izdatel Nauka, 1991.

[58] L. M. Kachanov. On the creep fracture time. *Izv. Akad. Nauk USSR*, 8:26–31, 1958.

[59] N. R. Hansen and H. L. Schreyer. A thermodynamically consistent framework for theories of elastoplasticity coupled with damage. *International Journal of Solids and Structures*, 31(3):359–389, 1994.

[60] J. Janson and J. Hult. Fracture mechanics and damage mechanics - a combined approach. *J. Mec. Theor. Appl. 1*, pages S18–28, 1977.

[61] E. J. Barbero and K. W. Kelly. Predicting high temperature ultimate strength of continuous fiber metal matrix composites. *Journal of Composite Materials*, 27(12):1214–1235, 1993.

[62] K. W. Kelly and E. Barbero. Effect of fiber damage on the longitudinal creep of a cfmmc. *International Journal of Solids and Structures*, 30(24):3417–3429, 1993.

[63] J. R. Rice. *Continuum Mechanics and Thermodynamics of Plasticity in Relation to Microscale Deformation Mechanisms*, chapter 2, pages 23–79. Constitutive Equations in Plasticity. MIT Press, Cambridge, MA, 1975.

[64] J. C. Simo and T. J. R. Hughes. *Computational Inelasticity.* Springer, Berlin, 1998.

[65] D. Krajcinovic, J. Trafimow, and D. Sumarac. Simple constitutive model for a cortical bone. *Journal of Biomechanics*, 20(8):779–784, 1987.

[66] D. Krajcinovic and D. Fanella. Micromechanical damage model for concrete. *Engineering Fracture Mechanics*, 25(5-6):585–596, 1985.

[67] D. Krajcinovic. Damage mechanics. *Mechanics of Materials*, 8(2-3):117–197, 1989.

[68] A. M. Neville. *Properties of Concrete, 2nd Edition.* Wiley, New York, NY, 1973.

[69] ACI, American Concrete Institute.

[70] B. W. Rosen. The tensile failure of fibrous composites. *AIAA Journal*, 2(11):1985–1911, 1964.

[71] W. Weibull. A statistical distribution function of wide applicability. *Journal of Applied Mechanics*, 18:293–296, 1951.

[72] B. W. Rosen. *Fiber Composite Materials*, chapter 3. American Society for Metals, Metals Park, OH, 1965.

[73] A. S. D. Wang. A non-linear microbuckling model predicting the compressive strength of unidirectional composites. In *ASME Winter Annual Meeting*, volume WA/Aero-1, 1978.

[74] J. S. Tomblin, E. J. Barbero, and L. A. Godoy. Imperfection sensitivity of fiber micro-buckling in elastic-nonlinear polymer-matrix composites. *International Journal of Solids and Structures*, 34(13):1667–1679, 1997.

[75] D. C. Lagoudas and A. M. Saleh. Compressive failure due to kinking of fibrous composites. *Journal of Composite Materials*, 27(1):83–106, 1993.

[76] P. Steif. A model for kinking in fiber composites. i. fiber breakage via micro-buckling. *International Journal of Solids and Structures*, 26 (5-6):549–61, 1990.

[77] S. W. Yurgartis and S. S. Sternstein. *Experiments to Reveal the Role of Matrix Properties and Composite Microstructure in Longitudinal Compression Strength*, volume 1185 of *ASTM Special Technical Publication*, pages 193–204. ASTM, Philadelphia, PA, 1994.

[78] C. Sun and A. W. Jun. Effect of matrix nonlinear behavior on the compressive strength of fiber composites. In *AMD*, volume 162, pages 91–101, New York, NY, 1993. ASME, American Society of Mechanical Engineers, Applied Mechanics Division.

[79] D. Adams and E. Lewis. Current status of composite material shear test methods. *SAMPE Journal*, 31(1):32–41, 1995.

[80] A. Maewal. Postbuckling behavior of a periodically laminated medium in compression. *International Journal of Solids and Structures*, 17(3):335–344, 1981.

[81] L. M. Kachanov. Rupture time under creep conditions problems of continuum mechanics. *SIAM*, pages 202–218, 1958.

[82] E. J. Barbero and P. Lonetti. Damage model for composites defined in terms of available data. *Mechanics of Composite Materials and Structures*, 8(4):299–315, 2001.

[83] E. J. Barbero and P. Lonetti. An inelastic damage model for fiber reinforced laminates. *Journal of Composite Materials*, 36(8):941–962, 2002.

[84] P. Lonetti, R. Zinno, F. Greco, and E. J. Barbero. Interlaminar damage model for polymer matrix composites. *Journal of Composite Materials*, 37(16):1485–1504, 2003.

[85] P. Lonetti, E. J. Barbero, R. Zinno, and F. Greco. Erratum: Interlaminar damage model for polymer matrix composites. *Journal of Composite Materials*, 38(9):799–800, 2004.

[86] E. J. Barbero, F. Greco, and P. Lonetti. Continuum damage-healing mechanics with application to self-healing composites. *International Journal of Damage Mechanics*, 14(1):51–81, 2005.

[87] S. Murakami. Mechanical modeling of material damage. *Journal of Applied Mechanics*, 55:281–286, 1988.

[88] J. M. Smith and H. C. Van Ness. *Introduction to Chemical Engineering Thermodynamics, 3rd Edition.* McGraw-Hill, New York, NY, 1975.

[89] E. R. Cohen, T. Cvita, J. G. Frey, B. Holmstrm, K. Kuchitsu, R. Marquardt, I. Mills, F. Pavese, M. Quack, J. Stohner, H. L. Strauss, M. Takami, and A. J. Thor. *Quantities, Units and Symbols in Physical Chemistry : The IUPAC Green Book, 3rd Edition, http://old.iupac.org/publications/books/author/cohen.html.* RSC Publishing, 2007.

[90] L. E. Malvern. *Introduction to the Mechanics of a Continuous Medium.* Prentice Hall, Upper Saddle River, NJ, 1969.

[91] Y. A. Cengel and M. A. Boles. *Thermodynamics: An Engineering Approach, 3rd Edition.* McGraw-Hill, New York, NY, 1998.

[92] J. Lubliner. *Plasticity Theory.* Collier Macmillan, New York, NY, 1990.

[93] J. A. Nairn. The strain energy release rate of composite microcracking: a variational approach. *Journal of Composite Materials*, 23(11):1106–29, 11 1989.

[94] J. A. Nairn, S. Hu, S. Liu, and J. S. Bark. The initiation, propagation, and effect of matrix microcracks in cross-ply and related laminates. In *Proc. of the 1st NASA Advanced Comp. Tech. Conf.*, pages 497–512, Oct. 29 - Nov. 1, 1990.

[95] S. Liu and J. A. Nairn. The formation and propagation of matrix microcracks in cross-ply laminates during static loading. *Journal of Reinforced Plastics and Composites*, 11(2):158–78, Feb. 1992.

[96] J. A. Nairn. Microcracking, microcrack-induced delamination, and longitudinal splitting of advanced composite structures. Technical Report NASA Contractor Report 4472, 1992.

[97] J. A. Nairn and S. Hu. *Damage Mechanics of Composite Materials*, volume 9 of *Composite Materials Series*, chapter Matrix Microcracking, pages 187–244. Elsevier, 1994.

[98] J. A. Nairn. Applications of finite fracture mechanics for predicting fracture events in composites. In *5th Int'l Conf. on Deform. and Fract. of Comp.*, 18-19 March 1999.

[99] J. A. Nairn. *Polymer Matrix Composites*, volume 2 of *Comprehensive Composite Materials*, chapter Matrix Microcracking in Composites, pages 403–432. Elsevier Science, 2000.

[100] J. A. Nairn. Fracture mechanics of composites with residual stresses, imperfect interfaces, and traction-loaded cracks. In *Workshop 'Recent Advances in Continuum Damage Mechanics for Composites'*, volume 61, pages 2159–67, UK, 20-22 Sept. 2000. Elsevier.

[101] J. A. Nairn and D. A. Mendels. On the use of planar shear-lag methods for stress-transfer analysis of multilayered composites. *Mechanics of Materials*, 33(6):335–362, 2001.

[102] J. A. Nairn. *Finite Fracture Mechanics of Matrix Microcracking in Composites*, pages 207–212. Application of Fracture Mechanics to Polymers, Adhesives and Composites. Elsevier, 2004.

[103] Y. M. Han, H. T. Hahn, and R. B. Croman. A simplified analysis of transverse ply cracking in cross-ply laminates. *Composites Science and Technology*, 31(3):165–177, 1988.

[104] C. T. Herakovich, J. Aboudi, S. W. Lee, and E. A. Strauss. Damage in composite laminates: Effects of transverse cracks. *Mechanics of Materials*, 7(2):91–107, 11 1988.

[105] J. Aboudi, S. W. Lee, and C. T. Herakovich. Three-dimensional analysis of laminates with cross cracks. *Journal of Applied Mechanics, Transactions ASME*, 55(2):389–397, 1988.

[106] F. W. Crossman, W. J. Warren, A. S. D. Wang, and G. E. Law. Initiation and growth of transverse cracks and edge delamination in composite laminates. ii - experimental correlation. *Journal of Composite Materials Suplement*, 14(1):88–108, 1980.

[107] D. L. Flaggs and M. H. Kural. Experimental determination of the in situ transverse lamina strength in graphite/epoxy laminates. *Journal of Composite Materials*, 16:103–16, 03 1982.

[108] S. H. Lim and S. Li. Energy release rates for transverse cracking and delaminations induced by transverse cracks in laminated composites. *Composites Part A*, 36(11):1467–1476, 2005.

[109] S. Li and F. Hafeez. Variation-based cracked laminate analysis revisited and fundamentally extended. *International Journal of Solids and Structures*, 46(20):3505–3515, 2009.

[110] J. L Rebiere and D. Gamby. A decomposition of the strain energy release rate associated with the initiation of transverse cracking, longitudinal cracking and delamination in cross-ply laminates. *Composite Structures*, 84(2):186–197, 2008.

[111] S. C. Tan and R. J. Nuismer. A theory for progressive matrix cracking in composite laminates. *Journal of Composite Materials*, 23:1029–1047, 1989.

[112] R. J. Nuismer and S. C. Tan. Constitutive relations of a cracked composite lamina. *Journal of Composite Materials*, 22:306–321, 1988.

[113] T. Yokozeki and T. Aoki. Stress analysis of symmetric laminates with obliquely-crossed matrix cracks. *Advanced Composite Materials*, 13(2):121–40, 2004.

[114] T. Yokozeki and T. Aoki. Overall thermoelastic properties of symmetric laminates containing obliquely crossed matrix cracks. *Composites Science and Technology*, 65(11-12):1647–54, 2005.

[115] Z. Hashin. Analysis of cracked laminates: a variational approach. *Mechanics of Materials*, 4:121:136, 1985.

[116] J. Zhang, J. Fan, and C. Soutis. Analysis of multiple matrix cracking in $[\pm\theta/90_n]_s$ composite laminates, part I, inplane stiffness properties. *Composites*, 23(5):291–304, 1992.

[117] P. Gudmundson and S. Ostlund. First order analysis of stiffness reduction due to matrix cracking. *Journal of Composite Materials*, 26(7):1009–30, 1992.

[118] P. Gudmundson and S. Ostlund. Prediction of thermoelastic properties of composite laminates with matrix cracks. *Composites Science and Technology*, 44(2):95–105, 1992.

[119] P. Gudmundson and W. Zang. An analytic model for thermoelastic properties of composite laminates containing transverse matrix cracks. *International Journal of Solids and Structures*, 30(23):3211–31, 1993.

[120] E. Adolfsson and P. Gudmundson. Thermoelastic properties in combined bending and extension of thin composite laminates with transverse matrix cracks. *International Journal of Solids and Structures*, 34(16):2035–60, 06 1997.

[121] E. Adolfsson and P. Gudmundson. Matrix crack initiation and progression in composite laminates subjected to bending and extension. *International Journal of Solids and Structures*, 36(21):3131–3169, 1999.

[122] W. Zang and P. Gudmundson. Damage evolution and thermoelastic properties of composite laminates. *International Journal of Damage Mechanics*, 2(3):290–308, 07 1993.

[123] E. Adolfsson and P. Gudmundson. Matrix crack induced stiffness reductions in [(0m/90n/+p/-q)s]m composite laminates. *Composites Engineering*, 5(1):107–23, 1995.

[124] P. Lundmark and J. Varna. Modeling thermo-mechanical properties of damaged laminates. In *3rd International Conference on Fracture and Damage Mechanics, FDM 2003*, volume 251-252 of *Advances in Fracture and Damage Mechanics*, pages 381–7, Switzerland, 2-4 Sept. 2003. Trans Tech Publications.

[125] M. Kachanov. *Elastic solids with many cracks and related problems*, volume 30 of *Advances in Applied Mechanics*, chapter X, pages 260–445. Academic Press, Inc., 1993.

[126] A. Adumitroiaie and E. J. Barbero. Intralaminar damage model for laminates subjected to membrane and flexural deformations. *Mechanics of Advanced Materials and Structures*, 2013.

[127] T. Yokozeki, T. Aoki, and T. Ishikawa. Transverse crack propagation in the specimen width direction of cfrp laminates under static tensile loadings. *Journal of Composite Materials*, 36(17):2085–99, 2002.

[128] T. Yokozeki, T. Aoki, and T. Ishikawa. Consecutive matrix cracking in contiguous plies of composite laminates. *International Journal of Solids and Structures*, 42(9-10):2785–802, 05 2005.

[129] T. Yokozeki, T. Aoki, T. Ogasawara, and T. Ishikawa. Effects of layup angle and ply thickness on matrix crack interaction in contiguous plies of composite laminates. *Composites Part A (Applied Science and Manufacturing)*, 36(9):1229–35, 2005.

[130] E. J. Barbero, F. A. Cosso, and F. A. Campo. Benchmark solution for degradation of elastic properties due to transverse matrix cracking in laminated composites. *Composite Structures*, 98(4):242–252, 2013.

[131] S. Li, S. R. Reid, and P. D. Soden. A continuum damage model for transverse matrix cracking in laminated fibre-reinforced composites. *Philosophical Transactions of the Royal Society London, Series A (Mathematical, Physical and Engineering Sciences)*, 356(1746):2379–412, 10/15 1998.

[132] Janis Varna, Roberts Joffe, and Ramesh Talreja. A synergistic damage-mechanics analysis of transverse cracking $[\pm\theta/90_4]s$ laminates. *Composites Science and Technology*, 61(5):657–665, 2001.

[133] A. S. D. Wang, P. C. Chou, and S. C. Lei. A stochastic model for the growth of matrix cracks in composite laminates. *Journal of Composite Materials*, 18(3):239–54, 05 1984.

[134] S. Li, S. R. Reid, and P. D. Soden. Modelling the damage due to transverse matrix cracking in fiber-reinforced laminates. In *Proc. 2nd Int. Conf. on Nonlinear Mechanics (ICNP-2)*, pages 320–323. Peking University Press, 1993.

[135] D.H. Cortes and E.J. Barbero. Stiffness reduction and fracture evolution of oblique matrix cracks in composite laminates. *Annals of Solid and Structural Mechanics*, 1(1):29–40, 2010.

[136] E. J. Barbero and D. H. Cortes. A mechanistic model for transverse damage initiation, evolution, and stiffness reduction in laminated composites. *Composites Part B*, 41:124–132, 2010.

[137] E. J. Barbero, G. Sgambitterra, A. Adumitroaie, and X. Martinez. A discrete constitutive model for transverse and shear damage of symmetric laminates with arbitrary stacking sequence. *Composite Structures*, 93:1021–1030, 2011.

[138] G. Sgambitterra, A. Adumitroaie, E.J. Barbero, and A. Tessler. A robust three-node shell element for laminated composites with matrix damage. *Composites Part B: Engineering*, 42(1):41–50, 2011.

[139] A. M. Abad Blazquez, M. Herraez Matesanz, Carlos Navarro Ugena, and E. J. Barbero. Acoustic emission characterization of intralaminar damage in composite laminates. In *Asociación Española de Materiales Compuestos MATCOMP 2013*, Algeciras, Spain, July 2–5, 2013.

[140] E. J. Barbero and F. A. Cosso. Determination of material parameters for discrete damage mechanics analysis of composite laminates. *Composites Part B:Engineering*, 2013.

[141] H. T. Hahn. A mixed-mode fracture criterion for composite materials. *Composites Technology Review*, 5:26–29, 1983.

[142] Allan F. Bower. *Mechanics of Solids*. CRC, 2010. Free at http://solidmechanics.org/.

[143] Simulia. Abaqus analysis user's manual, version 6.10.

[144] J. Varna, R. Joffe, N. V. Akshantala, and R. Talreja. Damage in composite laminates with off-axis plies. *Composites Science and Technology*, 59(14):2139–2147, 1999.

[145] H. Chai, C. D. Babcock, and W. G. Knauss. One dimensional modeling of failure in laminated plates by delamination buckling. *International Journal of Solids and Structures*, 17(11):1069–1083, 1981.

[146] M.-K. Yeh and L.-B. Fang. Contact analysis and experiment of delaminated cantilever composite beam. *Composites: Part B*, 30(4):407–414, 1999.

[147] W. L. Yin, S. N. Sallam, and G. J. Simitses. Ultimate axial load capacity of a delaminated beam-plate. *AIAA Journal*, 24(1):123–128, 1986.

[148] D. Bruno. Delamination buckling in composite laminates with interlaminar defects. *Theoretical and Applied Fracture Mechanics*, 9(2):145–159, 1988.

[149] D. Bruno and A. Grimaldi. Delamination failure of layered composite plates loaded in compression. *International Journal of Solids and Structures*, 26(3):313–330, 1990.

[150] G. A. Kardomateas. The initial post-buckling and growth behaviour of internal delaminations in composite plates. *Journal of Applied Mechanics*, 60(4):903–910, 1993.

[151] G. A. Kardomateas and A. A. Pelegri. The stability of delamination growth in compressively loaded composite plates. *International Journal of Fracture*, 65(3):261–276, 1994.

[152] I. Sheinman, G. A. Kardomateas, and A. A. Pelegri. Delamination growth during pre- and post-buckling phases of delaminated composite laminates. *International Journal of Solids and Structures*, 35(1-2):19–31, 1998.

[153] D. Bruno and F. Greco. An asymptotic analysis of delamination buckling and growth in layered plates. *International Journal of Solids and Structures*, 37(43):6239–6276, 2000.

[154] H.-J. Kim. Postbuckling analysis of composite laminates with a delamination. *Computer and Structures*, 62(6):975–983, 1997.

[155] W. G. Bottega and A. Maewal. Delamination buckling and growth in laminates–closure. *Journal of Applied Mechanics*, 50(14):184–189, 1983.

[156] B. Cochelin and M. Potier-Ferry. A numerical model for buckling and growth of delaminations in composite laminates. *Computer Methods in Applied Mechanics*, 89(1-3):361–380, 1991.

[157] P.-L. Larsson. On delamination buckling and growth in circular and annular orthotropic plates. *International Journal of Solids and Structures*, 27(1):15–28, 1991.

[158] W.-L. Yin. Axisymmetric buckling and growth of a circular delamination in a compressed laminate. *International Journal of Solids and Structures*, 21(5):503–514, 1985.

[159] K.-F. Nilsson, L. E. Asp, J. E. Alpman, and L. Nystedt. Delamination buckling and growth for delaminations at different depths in a slender composite panel. *International Journal of Solids and Structures*, 38(17):3039–3071, 2001.

[160] H. Chai and C. D. Babcock. Two-dimensional modeling of compressive failure in delaminated laminates. *Journal of Composite Materials*, 19(1):67–98, 1985.

[161] J. D. Withcomb and K. N. Shivakumar. Strain-energy release rate analysis of plates with postbuckled delaminations. *Journal of Composite Materials*, 23(7):714–734, 1989.

[162] W. J. Bottega. A growth law for propagation of arbitrary shaped delaminations in layered plates. *International Journal of Solids and Structures*, 19(11):1009–1017.

[163] B. Storåkers and B. Andersson. Nonlinear plate theory applied to delamination in composites. *Journal of the Mechanics and Physics of Solids*, 36(6):689–718, 1988.

[164] P.-L. Larsson. On multiple delamination buckling and growth in composite plates. *International Journal of Solids and Structures*, 27(13):1623–1637, 1991.

[165] M. A. Kouchakzadeh and H. Sekine. Compressive buckling analysis of rectangular composite laminates containing multiple delaminations. *Composite Structures*, 50(3):249–255, 2000.

[166] J. M. Comiez, A. M. Waas, and K. W. Shahwan. Delamination buckling: Experiment and analysis. *International Journal of Solids and Structures*, 32(6/7):767–782, 1995.

[167] G. A. Kardomateas. Postbuckling characteristics in delaminated kevlar/epoxy laminates: An experimental study. *Journal of Composites Technology and Research*, 12(2):85–90, 1990.

[168] O. Allix and A. Corigliano. Modeling and simulation of crack propagation in mixed-modes interlaminar fracture specimens. *International Journal of Fracture*, 77(2):111–140, 1996.

[169] D. Bruno and F. Greco. Delamination in composite plates: Influence of shear deformability on interfacial debonding. *Cement and Concrete Composites*, 23(1):33–45, 2001.

[170] J. W. Hutchinson and Z. Suo. Mixed mode cracking in layered materials. *Advances in Applied Mechanics*, 29:63–191, 1992.

[171] N. Point and E. Sacco. Delamination of beams: An application to the dcb specimen. *International Journal of Fracture*, 79(3):225–247, 1996.

[172] J. G. Williams. On the calculation of energy release rate for cracked laminates. *International Journal of Fracture*, 36(2):101–119, 1988.

[173] A. Tylikowsky. Effects of piezoactuator delamination on the transfer functions of vibration control systems. *International Journal of Solids and Structures*, 38(10-13):2189–2202, 2001.

[174] C. E. Seeley and A. Chattopadhyay. Modeling of adaptive composites including debonding. *International Journal Solids and Structures*, 36(12):1823–1843, 1999.

[175] W. J. Bottega and A. Maewal. Dynamics of delamination buckling. *International Journal of Non-Linear Mechanics*, 18(6):449–463, 1983.

[176] G. Alfano and M. A. Crisfield. Finite element interface models for the delamination analysis of laminated composites: Mechanical and computational issues. *International Journal for Numerical Methods in Engineering*, 50(7):1701–1736, 2001.

[177] P. P. Camanho, C. G. Dávila, and M. F. Moura. Numerical simulation of mixed-mode progressive delamination in composite materials. *Journal of Composite Materials*, 37(16):1415–1438, 2003.

[178] Z. Zou, S. R. Reid, and S. Li. A continuum damage model for delaminations in laminated composites. *Journal of the Mechanics and Physics of Solids*, 51(2):333–356, 2003.

[179] X-P Xu and A. Needleman. Numerical simulations of fast crack growth in brittle solids. *Journal of the Mechanics and Physics of Solids*, 42:1397–1434, 1994.

[180] J. R. Reeder. A bilinear failure criterion for mixed-mode delamination in composite materials. *ASTM STP 1206*, pages 303–322, 1993.

[181] H. Chai. Three-dimensional fracture analysis of thin-film debonding. *International Journal of Fracture*, 46(4):237–256, 1990.

[182] J. D. Withcomb. Three dimensional analysis of a post-buckled embedded delamination. *Journal of Composite Materials*, 23(9):862–889, 1989.

[183] D. Bruno, F. Greco, and P. Lonetti. A 3d delamination modelling technique based on plate and interface theories for laminated structures. *European Journal of Mechanics A/Solids*, 24:127–149, 2005.

[184] J. G. Williams. The fracture mechanics of delamination tests. *Journal of Strain Analysis*, 24(4):207–214, 1989.

[185] W.-L. Yin and J. T. S. Wang. The energy-release rate in the growth of a one-dimensional delamination. *Journal of Applied Mechanics*, 51(4):939–941, 1984.

[186] J. R. Rice. A path independent integral and the approximate analysis of strain concentrations by notches and cracks. *Journal of Applied Mechanics*, 35:379–386, 1968.

[187] Simulia. Abaqus analysis user's manual, version 6.10. chapter 29.5: Cohesive elements.

[188] C. G. Dávila, P. P. Camanho, and M. F. De Moura. Mixed-mode decohesion elements for analyses of progressive delamination. In *42nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, April 16 - 19, 2001*, volume 3, pages 2277– 88, Seattle, WA, 2001. AIAA.

[189] Simulia. Abaqus analysis user's manual, version 6.10. chapter 11.4.3: Crack propagation analysis.

[190] E.J. Barbero and J.N. Reddy. Jacobian derivative method for three-dimensional fracture mechanics. *Communications in Applied Numerical Methods*, 6(7):507–518, 1990.

[191] M. Benzeggagh and M. Kenane. Measurement of mixed-mode delamination fracture toughness of unidirectional glass/epoxy composites with mixed-mode bending apparatus. *Composite Science and Technology*, 56:439, 1996.

[192] J. Reeder, S. Kyongchan, P. B. Chunchu, and D. R.. Ambur. Postbuckling and growth of delaminations in composite plates subjected to axial compression. In *43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Denver, Colorado, vol. 1746, p. 10*, 2002.

[193] A. Caceres. *Local Damage Analysis of Fiber Reinforced Polymer Matrix Composites, Ph.D. dissertation.* PhD thesis, West Virginia University, Morgantown, WV, 1998.

[194] E. J. Barbero. Composite materials forum. `http://forum.cadec-online.com/viewforum.php?f=4`.

[195] G. Puri. *Python Scripts for Abaqus.* abaquspython.org, Atlanta, GA, 2011.

[196] M. Metcalf. A tool to convert f77 source code to f90. `ftp://ftp.numerical.rl.ac.uk/pub/MandR/convert.f90`.

MATERIALS SCIENCE/MECHANICAL ENGINEERING

Finite Element Analysis of Composite Materials Using Abaqus™

"In my opinion, the book is very well written; it is easy to follow and includes topics that students, engineers, and researchers from different fields can find very interesting and useful. The examples are very well detailed and provide valuable guidance on how to implement and understand theoretical solutions when translated into finite element models of composite materials and structures. Overall, this is a great book if the reader is looking for a comprehensive treatment of composite materials covering detailed theoretical background followed by implementation of the concepts into practical problems using a powerful modeling tool such as Abaqus."

—EDUARDO M. SOSA, West Virginia University

"The book is essential for any academic in the area of solid mechanics. I use this book with my students, as the subject and the materials are very clear. Model files for the finite element examples in the book help students progress and make my guidance more productive. It has both theory and applications using the finite element method. This book is also essential for composite engineers as a quick reference of topics that can be of use in their field."

—GASSER ABDELAL, Queen's University Belfast

"This book by Professor Barbero does an excellent job introducing the fundamentals of mechanics of composite materials and the finite element method in a concise way. Some of the most common problems that the practicing engineer has to face when designing with composites using finite element analysis are covered in detail. ... a valuable asset for any reader dealing with modeling of composite structures using the finite element method."

—JOAQUIN GUTIERREZ, Blade Dynamics LLLP

CRC Press
Taylor & Francis Group
an informa business
www.taylorandfrancisgroup.com

6000 Broken Sound Parkway, NW
Suite 300, Boca Raton, FL 33487
711 Third Avenue
New York, NY 10017
2 Park Square, Milton Park
Abingdon, Oxon OX14 4RN, UK

K15072
ISBN: 978-1-4665-1661-8
90000
9 781466 516618
www.crcpress.com