Voyiadjis

Kattan

Mechanics of Composite Materials with MATLAB

Springer

EXTRA MATERIALS
extras.springer.com

Mechanics of Composite Materials with MATLAB

George Z. Voyiadjis   Peter I. Kattan

Mechanics of Composite Materials with MATLAB

With 86 Figures and a CD ROM

Springer

Prof. George Z. Voyiadjis
Prof. Peter I. Kattan
Louisiana State University
Dept.Civil and Environmental Engineering
Baton Rouge, LA 70803, USA
e-mail: voyiadjis@eng.lsu.edu
pkattan@lsu.edu

Library of Congress Control Number: 2005920509

ISBN-10 3-540-24353-4 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-24353-3 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media
springeronline.com

Printed in The Netherlands

Typesetting: by the authors and TechBooks using a Springer LaTeX macro package
Cover design: deblik, Berlin

Printed on acid-free paper SPIN: 11015482 89/3141/jl 5 4 3 2 1 0

Dedicated with Love to CHRISTINA, ELENA, and ANDREW
George Z. Voyiadjis

Dedicated with Love to My Family
Peter I. Kattan

Preface

This is a book for people who love mechanics of composite materials and MATLAB*. We will use the popular computer package MATLAB as a matrix calculator for doing the numerical calculations needed in mechanics of composite materials. In particular, the steps of the mechanical calculations will be emphasized in this book. The reader will not find ready-made MATLAB programs for use as black boxes. Instead step-by-step solutions of composite material mechanics problems are examined in detail using MATLAB. All the problems in the book assume linear elastic behavior in structural mechanics. The emphasis is not on mass computations or programming, but rather on learning the composite material mechanics computations and understanding of the underlying concepts.

The basic aspects of the mechanics of fiber-reinforced composite materials are covered in this book. This includes lamina analysis in both the local and global coordinate systems, laminate analysis, and failure theories of a lamina. In the last two chapters of the book, we present a glimpse into two especially advanced topics in this subject, namely, homogenization of composite materials, and damage mechanics of composite materials. The authors have deliberately left out the two topics of laminated plates and stability of composites as they feel these two topics are a little bit advanced for the scope of this book. In addition, each of these topics deserves a separate volume for its study and there are some books dedicated to these two topics. Each chapter starts with a summary of the basic equations. This is followed by the MATLAB functions which are specific to the chapter. Then, a number of examples is solved demonstrating both the theory and numerical computations. The examples are of two types: the first type is theoretical and involves derivations and proofs of various equations, while the other type is MATLAB-based and involves using MATLAB in the calculations. A total of 44 special MATLAB functions for composite material mechanics are provided as M-files on the accompanying CD-ROM to be used in the examples and solution of the

* MATLAB is a registered trademark of the MathWorks, Inc.

problems. These MATLAB functions are specifically written by the authors to be used with this book. These functions have been tested successfully with MATLAB versions 6.0 and 6.2. They should work with other later or previous versions. Each chapter also ends with a number of problems to be used as practice for students.

The book is written primarily for students studying mechanics of composite materials for the first time. The book is self-contained and can be used as a textbook for an introductory course on mechanics of composite materials. Since the computations of composite materials usually involve matrices and matrix manipulations, it is only natural that students use a matrix-based software package like MATLAB to do the calculations. In fact the word MATLAB stands for MATrix LABoratory.

The main features of this book are listed as follows:

1. The book is divided into twelve chapters that are well defined and correlated. Each chapter is written in a way to be consistent with the other chapters.
2. The book includes a short tutorial on using MATLAB in Chap. 1.
3. The CD-ROM that accompanies the book includes 44 MATLAB functions (M-files) that are specifically written by the authors to be used with this book. These functions comprise what may be called the MATLAB Composite Material Mechanics Toolbox. It is used mainly for problems in structural mechanics. The provided MATLAB functions are designed to be simple and easy to use.
4. The book stresses the interactive use of MATLAB. The MATLAB examples are solved in an interactive manner in the form of interactive sessions with MATLAB. No ready-made subroutines are provided to be used as black boxes. These latter ones are available in other books and on the internet.
5. Some of the examples show in detail the derivations and proofs of various basic equations in the study of the mechanics of composite materials. The derivations of the remaining equations are left to some of the problems.
6. Solutions to most of the problems are included in a special section at the end of the book. These solutions are detailed especially for the first six chapters.

The authors wish to thank the editors at Springer-Verlag (especially Dr. Thomas Ditzinger) for their cooperation and assistance during the writing of this book. Special thanks are also given to our family members without their support and encouragement this book would not have been possible. The second author would also like to acknowledge the financial support of the Center for Computation and Technology headed by Edward Seidel at Louisiana State University.

Louisiana State University
February 2005

George Z. Voyiadjis
Peter I. Kattan

Contents

1 Introduction

This short introductory chapter is divided into two parts. In the first part there is an overview of the mechanics of fiber-reinforced composite materials. The second part includes a short tutorial on MATLAB.

1.1 Mechanics of Composite Materials

There are many excellent textbooks available on mechanics of fiber-reinforced composite materials like those in [1–12]. Therefore this book will not present any theoretical formulations or derivations of mechanics of composite materials. Only the main equations are summarized for each chapter followed by examples. In addition only problems from linear elastic structural mechanics are used throughout the book.

The main subject of this book is the mechanics of fiber-reinforced composite materials. These materials are usually composed of brittle fibers and a ductile matrix. The geometry is in the form of a laminate which consists of several parallel layers where each layer is called a lamina. The advantage of this construction is that it gives the material more strength and less weight.

The mechanics of composite materials deals mainly with the analysis of stresses and strains in the laminate. This is usually performed by analyzing the stresses and strains in each lamina first. The results for all the laminas are then integrated over the length of the laminate to obtain the overall quantities. In this book, Chaps. 2–6 deal mainly with the analysis of stress and strain in one single lamina. This is performed in the local lamina coordinate system and also in the global laminate coordinate system. Laminate analysis is then discussed in Chaps. 7–9. The analysis of a lamina and a laminate in these first nine chapters are supplemented by numerous MATLAB examples demonstrating the theory in great detail. Each MATLAB example is conducted in the form of an interactive MATLAB session using the supplied MATLAB functions. Each chapter of the first nine chapters has a set of special MATLAB functions written specifically for each chapter. There are MATLAB functions for lamina analysis and for laminate analysis.

In Chap. 10, we illustrate the basic concepts of the major four failure theories of a single lamina. We do not illustrate the failure of a complete laminate because this mainly depends on which lamina fails first and so on. Finally, Chaps. 11 and 12 provide an introduction to the advanced topics of homogenization and damage mechanics in composite materials, respectively. These two topics are very important and are currently under extensive research efforts worldwide.

The analyses discussed in this book are limited to linear elastic composite materials. The reader who is interested in advanced topics like elasto-plastic composites, temperature effects, creep effects, viscoplasticity, composite plates and shells, dynamics and vibration of composites, etc. may refer to the widely available literature on these topics.

1.2 MATLAB Functions for Mechanics of Composite Materials

The CD-ROM accompanying this book includes 44 MATLAB functions (M-files) specifically written by the authors to be used for the analysis of fiber-reinforced composite materials with this book. They comprise what may be called the MATLAB Composite Materials Mechanics Toolbox. The following is a listing of all the functions available on the CD-ROM. The reader can refer to each chapter for specific usage details.

OrthotropicCompliance(E1, E2, E3, NU12, NU23, NU13, G12, G23, G13)
OrthotropicStiffness(E1, E2, E3, NU12, NU23, NU13, G12, G23, G13)
TransverselyIsotropicCompliance(E1, E2, NU12, NU23, G12)
TransverselyIsotropicStiffness(E1, E2, NU12, NU23, G12)
IsotropicCompliance(E, NU)
IsotropicStiffness(E, NU)

E1(Vf, E1f, Em)
NU12(Vf, NU12f, NUm)
E2(Vf, E2f, Em, Eta, NU12f, NU21f, NUm, E1f, p)
G12(Vf, G12f, Gm, EtaPrime, p)
Alpha1(Vf, E1f, Em, Alpha1f, Alpham)
Alpha2(Vf, Alpha2f, Alpham, E1, E1f, Em, NU1f, NUm, Alpha1f, p)
E2Modified(Vf, E2f, Em, Eta, NU12f, NU21f, NUm, E1f, p)

ReducedCompliance(E1, E2, NU12, G12)
ReducedStiffness(E1, E2, NU12, G12)
ReducedIsotropicCompliance(E, NU)
ReducedIsotropicStiffness(E, NU)
ReducedStiffness2(E1, E2, NU12, G12)
ReducedIsotropicStiffness2(E, NU)

T(theta)
Tinv(theta)
Sbar(S, theta)
Qbar(Q, theta)
Tinv2(theta)
Sbar2(S, T)
Qbar2(Q, T)

Ex(E1, E2, NU12, G12, theta)
NUxy(E1, E2, NU12, G12, theta)
Ey(E1, E2, NU21, G12, theta)
NUyx(E1, E2, NU21, G12, theta)
Gxy(E1, E2, NU12, G12, theta)
Etaxyx(Sbar)
*Etaxy*y(Sbar)
*Etax*xy(Sbar)
*Etay*xy(Sbar)

Strains(eps_xo, eps_yo, gam_xyo, kap_xo, kap_yo, kap_xyo, z)

Amatrix(A, Qbar, z1, z2)
Bmatrix(B, Qbar, z1, z2)
Dmatrix(D, Qbar, z1, z2)

Ebarx(A, H)
Ebary(A, H)
NUbarxy(A, H)
NUbaryx(A, H)
Gbarxy(A, H)

1.3 MATLAB Tutorial

In this section a very short MATLAB tutorial is provided. For more details consult the excellent books listed in [13–21] or the numerous freely available tutorials on the internet – see [22–29]. This tutorial is not comprehensive but describes the basic MATLAB commands that are used in this book.

In this tutorial it is assumed that you have started MATLAB on your system successfully and you are ready to type the commands at the MATLAB prompt (which is denoted by double arrows "≫"). Entering scalars and simple operations is easy as is shown in the examples below:

```
>> 2 * 3 + 7

ans =
    13
```

```
>> sin(45*pi/180)

ans =

    0.7071

>> x = 6

x =

    6

>> 5/sqrt(2 - x)

ans =

    0 - 2.5000i
```

Notice that the last result is a complex number. To suppress the output in MATLAB use a semicolon to end the command line as in the following examples. If the semicolon is not used then the output will be shown by MATLAB:

```
>> y = 35;
>> z = 7;
>> x = 3 * y + 4 * z;
>> w = 2 * y - 5 * z

w =

    35
```

MATLAB is case-sensitive, i.e. variables with lowercase letters are different than variables with uppercase letters. Consider the following examples using the variables x and X.

```
>> x = 1

x =

    1

>> X = 2

X =

    2
>> x
```

```
x =

    1

>> X

X =

    2
```

Use the help command to obtain help on any particular MATLAB command. The following example demonstrates the use of `help` to obtain help on the `det` command.

```
>> help det

 DET    Determinant.
    DET(X) is the determinant of the square matrix X.

    Use COND instead of DET to test for matrix singularity.

    See also COND.

 Overloaded methods
    help sym/det.m
```

The following examples show how to enter matrices and perform some simple matrix operations:

```
>> x = [1 4 7 ; 3 5 6 ; 1 3 8]

x =

     1     4     7
     3     5     6
     1     3     8

>> y = [1 ; 3 ; 0 ]

y =

     1
     3
     0

>> w = x * y
```

```
w =

    13
    18
    10
```

Let us now solve the following system of simultaneous algebraic equations:

$$\begin{bmatrix} 1 & 4 & 6 & -5 \\ 3 & 1 & 0 & -1 \\ 3 & 7 & 2 & 1 \\ 0 & 1 & 3 & 5 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{Bmatrix} = \begin{Bmatrix} 1 \\ -2 \\ 0 \\ 5 \end{Bmatrix} \tag{1.1}$$

We will use Gaussian elimination to solve the above system of equations. This is performed in MATLAB by using the backslash operator "\" as follows:

```
>> A = [1 4 6 -5 ; 3 1 0 -1 ; 3 7 2 1 ; 0 1 3 5]

A =

     1     4     6    -5
     3     1     0    -1
     3     7     2     1
     0     1     3     5

>> b = [1 ; -2 ; 0 ; 5]

b =

     1
    -2
     0
     5

>> x = A\b

x =

   -0.4444
   -0.1111
    0.7778
    0.5556
```

It is clear that the solution is $x_1 = -0.4444$, $x_2 = -0.1111$, $x_3 = 0.7778$, and $x_4 = 0.5556$. Alternatively, one can use the inverse matrix of A to obtain the same solution directly as follows:

```
>> x = inv(A) * b

x =

   -0.4444
   -0.1111
    0.7778
    0.5556
```

It should be noted that using the inverse method usually takes longer than using Gaussian elimination especially for large systems.

Finally in order to plot a graph of the function $y = f(x)$, we use the MATLAB command `plot(x, y)` after we have adequately defined both vectors x and y. The following is a simple example.

```
>> x = [ 1 2 3 4 5 6 7 8 9 10]

x =

     1      2      3      4      5     6      7      8      9      10

>> y = x. ^ 3 - 2 * x. ^ 2 + 5

y =

     4      5     14     37     80   149    250    389    572     805
```

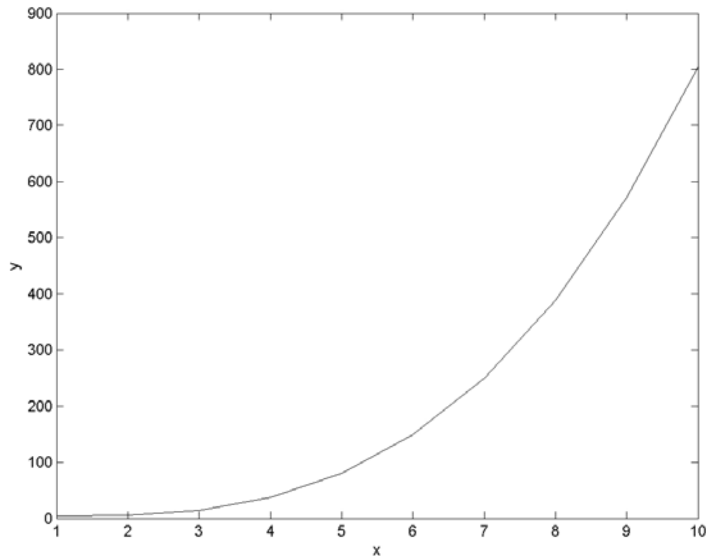

Fig. 1.1. Using the MATLAB `Plot` command

```
EDU >> plot(x, y)
EDU >> hold on;
EDU >> xlabel('x');
EDU >> ylabel('y');
```

Figure 1.1 shows the plot obtained by MATLAB. It is usually shown in a separate graphics window. Notice how the `xlabel` and `ylabel` MATLAB commands are used to label the two axes. Notice also how a "dot" is used in the function definition just before the exponentiation operation to indicate to MATLAB to carry out the operation on an element by element basis.

2

Linear Elastic Stress-Strain Relations

2.1 Basic Equations

Consider a single layer of fiber-reinforced composite material as shown in Fig. 2.1. In this layer, the 1-2-3 orthogonal coordinate system is used where the directions are taken as follows:

1. The 1-axis is aligned with the fiber direction.
2. The 2-axis is in the plane of the layer and perpendicular to the fibers.
3. The 3-axis is perpendicular to the plane of the layer and thus also perpendicular to the fibers.

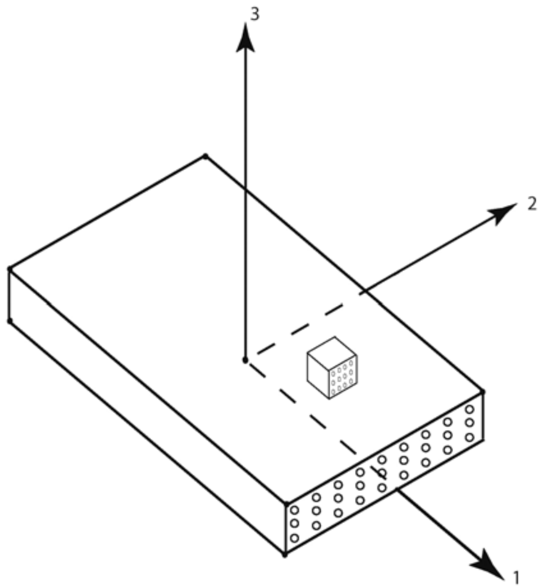

Fig. 2.1. A lamina illustrating the principle material coordinate system

The 1-direction is also called the *fiber direction*, while the 2- and 3-directions are called the *matrix directions* or the *transverse directions.* This 1-2-3 coordinate system is called the *principal material coordinate system.* The stresses and strains in the layer (also called a lamina) will be referred to the principal material coordinate system.

At this level of analysis, the strain or stress of an individual fiber or an element of matrix is not considered. The effect of the fiber reinforcement is smeared over the volume of the material. We assume that the two-material fiber-matrix system is replaced by a single homogeneous material. Obviously, this single material does not have the same properties in all directions. Such material with different properties in three mutually perpendicular directions is called an *orthotropic* material. Therefore, the layer (lamina) is considered to be orthotropic.

The stresses on a small infinitesimal element taken from the layer are illustrated in Fig. 2.2. There are three normal stresses σ_1, σ_2, and σ_3, and three shear stresses τ_{12}, τ_{23}, and τ_{13}. These stresses are related to the strains ε_1, ε_2, ε_3, γ_{12}, γ_{23}, and γ_{13} as follows (see [1]):

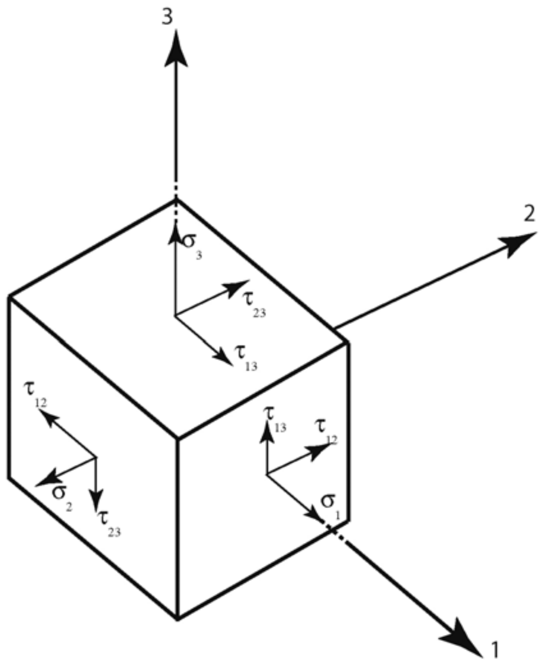

Fig. 2.2. An infinitesimal fiber-reinforced element showing the stresses

$$\begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \gamma_{23} \\ \gamma_{13} \\ \gamma_{12} \end{Bmatrix} = \begin{bmatrix} 1/E_1 & -\nu_{21}/E_2 & -\nu_{31}/E_3 & 0 & 0 & 0 \\ -\nu_{12}/E_1 & 1/E_2 & -\nu_{32}/E_3 & 0 & 0 & 0 \\ -\nu_{13}/E_1 & -\nu_{23}/E_2 & 1/E_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/G_{23} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/G_{13} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/G_{12} \end{bmatrix} \begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \tau_{23} \\ \tau_{13} \\ \tau_{12} \end{Bmatrix} \tag{2.1}$$

In (2.1), E_1, E_2, and E_3 are the extensional moduli of elasticity along the 1, 2, and 3 directions, respectively. Also, ν_{ij} $(i, j = 1,\ 2,\ 3)$ are the different Poisson's ratios, while G_{12}, G_{23}, and G_{13} are the three shear moduli.

Equation (2.1) can be written in a compact form as follows:

$$\{\varepsilon\} = [S]\,\{\sigma\} \tag{2.2}$$

where $\{\varepsilon\}$ and $\{\sigma\}$ represent the 6×1 strain and stress vectors, respectively, and $[S]$ is called the *compliance matrix*. The elements of $[S]$ are clearly obtained from (2.1), i.e. $S_{11} = 1/E_1$, $S_{12} = -\nu_{21}/E_2, \ldots, S_{66} = 1/G_{12}$.

The inverse of the compliance matrix $[S]$ is called the stiffness matrix $[C]$ given, in general, as follows:

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \tau_{23} \\ \tau_{13} \\ \tau_{12} \end{Bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & 0 & 0 & 0 \\ C_{21} & C_{22} & C_{23} & 0 & 0 & 0 \\ C_{31} & C_{32} & C_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \gamma_{23} \\ \gamma_{13} \\ \gamma_{12} \end{Bmatrix} \tag{2.3}$$

In compact form (2.3) is written as follows:

$$\{\sigma\} = [C]\,\{\varepsilon\} \tag{2.4}$$

The elements of $[C]$ are not shown here explicitly but are calculated using the MATLAB function *OrthotropicStiffness* which is written specifically for this purpose.

It is shown (see [1]) that both the compliance matrix and the stiffness matrix are symmetric, i.e. $C_{21} = C_{12}$, $C_{23} = C_{32}$, $C_{13} = C_{31}$, and similarly for S_{21}, S_{23}, and S_{13}. Therefore, the following expressions can now be easily obtained:

$$\begin{aligned} C_{11} &= \frac{1}{S}(S_{22}S_{33} - S_{23}S_{23}) \\ C_{12} &= \frac{1}{S}(S_{13}S_{23} - S_{12}S_{33}) \\ C_{22} &= \frac{1}{S}(S_{33}S_{11} - S_{13}S_{13}) \\ C_{13} &= \frac{1}{S}(S_{12}S_{23} - S_{13}S_{22}) \end{aligned}$$

$$
\begin{aligned}
C_{33} &= \frac{1}{S}(S_{11}S_{22} - S_{12}S_{12}) \\
C_{23} &= \frac{1}{S}(S_{12}S_{13} - S_{23}S_{11}) \\
C_{44} &= \frac{1}{S_{44}} \\
C_{55} &= \frac{1}{S_{55}} \\
C_{66} &= \frac{1}{S_{66}} \\
S &= S_{11}S_{22}S_{33} - S_{11}S_{23}S_{23} - S_{22}S_{13}S_{13} - S_{33}S_{12}S_{12} + 2S_{12}S_{23}S_{13}
\end{aligned}
\tag{2.5}
$$

It should be noted that the material constants appearing in the compliance matrix in (2.1) are not all independent. This is clear since the compliance matrix is symmetric. Therefore, we have the following equations relating the material constants:

$$
\begin{aligned}
\frac{\nu_{12}}{E_1} &= \frac{\nu_{21}}{E_2} \\
\frac{\nu_{13}}{E_1} &= \frac{\nu_{31}}{E_3} \\
\frac{\nu_{23}}{E_2} &= \frac{\nu_{32}}{E_3}
\end{aligned}
\tag{2.6}
$$

The above equations are called the *reciprocity relations* for the material constants. It should be noted that the reciprocity relations can be derived irrespective of the symmetry of the compliance matrix – in fact we conclude that the compliance matrix is symmetric from using these relations. Thus it is now clear that there are nine independent material constants for an orthotropic material.

A material is called *transversely isotropic* if its behavior in the 2-direction is identical to its behavior in the 3-direction. For this case, $E_2 = E_3$, $\nu_{12} = \nu_{13}$, and $G_{12} = G_{13}$. In addition, we have the following relation:

$$
G_{23} = \frac{E_2}{2(1 + \nu_{23})} \tag{2.7}
$$

It is clear that there are only five independent material constants (E_1, E_2, ν_{12}, ν_{23}, G_{12}) for a transversely isotropic material.

A material is called *isotropic* if its behavior is the same in all three 1-2-3 directions. In this case, $E_1 = E_2 = E_3 = E$, $\nu_{12} = \nu_{23} = \nu_{13} = \nu$, and $G_{12} = G_{23} = G_{13} = G$. In addition, we have the following relation:

$$
G = \frac{E}{2(1 + \nu)} \tag{2.8}
$$

It is clear that there are only two independent material constants (E, ν) for an isotropic material.

At the other end of the spectrum, we have *anisotropic* materials – these materials have nonzero entries at the upper right and lower left portions of their compliance and stiffness matrices.

2.2 MATLAB Functions Used

The six MATLAB functions used in this chapter to calculate compliance and stiffness matrices are:

OrthotropicCompliance(E1, E2, E3, NU12, NU23, NU13, G12, G23, G13) – This function calculates the 6×6 compliance matrix for orthotropic materials. Its input are the nine independent material constants E_1, E_2, E_3, ν_{12}, ν_{23}, ν_{13}, G_{12}, G_{23}, and G_{13}.

OrthotropicStiffness(E1, E2, E3, NU12, NU23, NU13, G12, G23, G13) – This function calculates the 6×6 stiffness matrix for orthotropic materials. Its input are the nine independent material constants E_1, E_2, E_3, ν_{12}, ν_{23}, ν_{13}, G_{12}, G_{23}, and G_{13}.

TransverselyIsotropicCompliance(E1, E2, NU12, NU23, G12) – This function calculates the 6×6 compliance matrix for transversely isotropic materials. Its input are the five independent material constants E_1, E_2, ν_{12}, ν_{23}, and G_{12}.

TransverselyIsotropicStiffness(E1, E2, NU12, NU23, G12) – This function calculates the 6×6 stiffness matrix for transversely isotropic materials. Its input are the five independent material constants E_1, E_2, ν_{12}, ν_{23}, and G_{12}.

IsotropicCompliance(E, NU) – This function calculates the 6×6 compliance matrix for isotropic materials. Its input are the two independent material constants E and ν.

IsotropicStiffness(E, NU) – This function calculates the 6×6 stiffness matrix for isotropic materials. Its input are the two independent material constants E and ν.

The following is a listing of the MATLAB source code for each function:

```
function y = OrthotropicCompliance(E1,E2,E3,NU12,NU23,NU13,G12,G23,G13)
%OrthotropicCompliance   This function returns the compliance matrix
%                        for orthotropic materials. There are nine
%                        arguments representing the nine independent
%                        material constants. The size of the compliance
%                        matrix is 6 x 6.
y = [1/E1 -NU12/E1 -NU13/E1 0 0 0 ; -NU12/E1 1/E2 -NU23/E2 0 0 0 ;
    -NU13/E1 -NU23/E2 1/E3 0 0 0 ; 0 0 0 1/G23 0 0 ; 0 0 0 0 1/G13 0 ;
    0 0 0 0 0 1/G12];
```

```
function y = OrthotropicStiffness(E1,E2,E3,NU12,NU23,NU13,G12,G23,G13)
%OrthotropicStiffness   This function returns the stiffness matrix
%                       for orthotropic materials. There are nine
%                       arguments representing the nine independent
%                       material constants. The size of the stiffness
%                       matrix is 6 x 6.
x = [1/E1 -NU12/E1 -NU13/E1 0 0 0 ; -NU12/E1 1/E2 -NU23/E2 0 0 0 ;
    -NU13/E1 -NU23/E2 1/E3 0 0 0 ; 0 0 0 1/G23 0 0 ; 0 0 0 0 1/G13 0 ;
    0 0 0 0 0 1/G12];
y = inv(x);
```

```
function y = TransverselyIsotropicCompliance(E1,E2,NU12,NU23,G12)
%TransverselyIsotropicCompliance   This function returns the
%                                  compliance matrix for
%                                  transversely isotropic
%                                  materials. There are five
%                                  arguments representing the
%                                  five independent material
%                                  constants. The size of the
%                                  compliance matrix is 6 x 6.
y = [1/E1 -NU12/E1 -NU12/E1 0 0 0 ; -NU12/E1 1/E2 -NU23/E2 0 0 0 ;
    -NU12/E1 -NU23/E2 1/E2 0 0 0 ; 0 0 0 2*(1+NU23)/E2 0 0 ;
    0 0 0 0 1/G12 0 ; 0 0 0 0 0 1/G12];
```

```
function y = TransverselyIsotropicStiffness(E1,E2,NU12,NU23,G12)
%TransverselyIsotropicStiffness   This function returns the
%                                 stiffness matrix for
%                                 transversely isotropic
%                                 materials. There are five
%                                 arguments representing the
%                                 five independent material
%                                 constants. The size of the
%                                 stiffness matrix is 6 x 6.
x = [1/E1 -NU12/E1 -NU12/E1 0 0 0 ; -NU12/E1 1/E2 -NU23/E2 0 0 0 ;
   -NU12/E1 -NU23/E2 1/E2 0 0 0 ; 0 0 0 2*(1+NU23)/E2 0 0 ;
    0 0 0 0 1/G12 0 ; 0 0 0 0 0 1/G12];
y = inv(x);
```

```
function y = IsotropicCompliance(E,NU)
%IsotropicCompliance   This function returns the
%                      compliance matrix for isotropic
%                      materials. There are two
%                      arguments representing the
%                      two independent material
%                      constants. The size of the
%                      compliance matrix is 6 x 6.
y = [1/E -NU/E -NU/E 0 0 0 ; -NU/E 1/E -NU/E 0 0 0 ;
    -NU/E -NU/E 1/E 0 0 0 ; 0 0 0 2*(1+NU)/E 0 0 ;
    0 0 0 0 2*(1+NU)/E 0 ; 0 0 0 0 0 2*(1+NU)/E];
```

```
function y = IsotropicStiffness(E,NU)
%IsotropicStiffness   This function returns the
%                     stiffness matrix for isotropic
%                     materials. There are two
%                     arguments representing the
%                     two independent material
%                     constants. The size of the
%                     stiffness matrix is 6 x 6.
x = [1/E -NU/E -NU/E 0 0 0 ; -NU/E 1/E -NU/E 0 0 0 ;
    -NU/E -NU/E 1/E 0 0 0 ; 0 0 0 2*(1+NU)/E 0 0 ;
    0 0 0 0 2*(1+NU)/E 0 ; 0 0 0 0 0 2*(1+NU)/E];
y = inv(x);
```

Example 2.1

For an orthotropic material, derive expressions for the elements of the stiffness matrix C_{ij} directly in terms of the nine independent material constants.

Solution

Substitute the elements of $[S]$ from (2.1) into (2.5) along with using (2.6). This is illustrated in detail for C_{11} below. First evaluate the expression of S from (2.5) as follows:

$$
\begin{aligned}
S = \; & S_{11}S_{22}S_{33} - S_{11}S_{23}S_{23} - S_{22}S_{13}S_{13} - S_{33}S_{12}S_{12} + 2S_{12}S_{23}S_{13} \\
= \; & \frac{1}{E_1}\frac{1}{E_2}\frac{1}{E_3} - \frac{1}{E_1}\left(\frac{-\nu_{23}}{E_2}\right)\left(\frac{-\nu_{32}}{E_3}\right) \\
& -\frac{1}{E_2}\left(\frac{-\nu_{13}}{E_1}\right)\left(\frac{-\nu_{31}}{E_3}\right) - \frac{1}{E_3}\left(\frac{-\nu_{12}}{E_1}\right)\left(\frac{-\nu_{21}}{E_2}\right) \\
& +2\left(\frac{-\nu_{12}}{E_1}\right)\left(\frac{-\nu_{23}}{E_2}\right)\left(\frac{-\nu_{31}}{E_3}\right) \\
= \; & \frac{1 - \nu_{23}\nu_{32} - \nu_{13}\nu_{31} - \nu_{12}\nu_{21} - 2\nu_{12}\nu_{23}\nu_{31}}{E_1 E_2 E_3} \\
= \; & \frac{1 - \nu_0}{E_1 E_2 E_3}
\end{aligned} \tag{2.9a}
$$

where ν_0 is given by

$$
\nu_0 = \nu_{23}\nu_{32} + \nu_{13}\nu_{31} + \nu_{12}\nu_{21} + 2\nu_{12}\nu_{23}\nu_{31} \tag{2.9b}
$$

Next, C_{11} is calculated as follows

$$\begin{aligned} C_{11} &= \frac{1}{S}(S_{22}S_{33} - S_{23}S_{23}) \\ &= \frac{E_1E_2E_3}{1-\nu_0}\left[\frac{1}{E_2}\frac{1}{E_3} - \left(\frac{-\nu_{23}}{E_2}\right)\left(\frac{-\nu_{32}}{E_3}\right)\right] \\ &= \frac{(1-\nu_{23}\nu_{32})\,E_1}{1-\nu_0} \end{aligned} \tag{2.9c}$$

Similarly, the following expressions for the other elements of $[C]$ can be derived:

$$C_{12} = \frac{(\nu_{21}+\nu_{31}\nu_{23})\,E_1}{1-\nu_0} = \frac{(\nu_{12}+\nu_{32}\nu_{13})\,E_2}{1-\nu_0} \tag{2.9d}$$

$$C_{13} = \frac{(\nu_{31}+\nu_{21}\nu_{32})\,E_1}{1-\nu_0} = \frac{(\nu_{13}+\nu_{12}\nu_{23})\,E_3}{1-\nu_0} \tag{2.9e}$$

$$C_{22} = \frac{(1-\nu_{13}\nu_{31})\,E_2}{1-\nu_0} \tag{2.9f}$$

$$C_{23} = \frac{(\nu_{32}+\nu_{12}\nu_{31})\,E_2}{1-\nu_0} = \frac{(\nu_{23}+\nu_{21}\nu_{13})\,E_3}{1-\nu_0} \tag{2.9g}$$

$$C_{33} = \frac{(1-\nu_{12}\nu_{21})\,E_3}{1-\nu_0} \tag{2.9h}$$

$$C_{44} = G_{23} \tag{2.9i}$$

$$C_{55} = G_{13} \tag{2.9j}$$

$$C_{66} = G_{12} \tag{2.9k}$$

MATLAB Example 2.2

Consider a 60-mm cube made of graphite-reinforced polymer composite material that is subjected to a tensile force of 100 kN perpendicular to the fiber direction, directed along the 2-direction. The cube is free to expand or contract. Use MATLAB to determine the changes in the 60-mm dimensions of the cube. The material constants for graphite-reinforced polymer composite material are given as follows [1]:

$$\begin{aligned} E_1 &= 155.0 \text{ GPa}, & E_2 &= E_3 = 12.10 \text{ GPa} \\ \nu_{23} &= 0.458, & \nu_{12} &= \nu_{13} = 0.248 \\ G_{23} &= 3.20 \text{ GPa}, & G_{12} &= G_{13} = 4.40 \text{ GPa} \end{aligned}$$

Solution

This example is solved using MATLAB. First, the normal stress in the 2-direction is calculated in GPa as follows:

```
>> sigma2 = 100/(60*60)

sigma2 =

    0.0278
```

The stress vector is set up next as follows:

```
>> sigma = [0  sigma2  0  0  0  0]

sigma =

    0  0.0278  0  0  0  0
```

The compliance matrix is then calculated using the MATLAB function *OrthotropicCompliance* as follows:

```
>> S = OrthotropicCompliance(155.0, 12.10, 12.10, 0.248, 0.458, 0.248,
4.40, 3.20, 4.40)

S =

    0.0065   -0.0016   -0.0016         0         0         0
   -0.0016    0.0826   -0.0379         0         0         0
   -0.0016   -0.0379    0.0826         0         0         0
         0         0         0    0.3125         0         0
         0         0         0         0    0.2273         0
         0         0         0         0         0    0.2273
```

The stress vector is adjusted to be a 6×1 column vector as follows:

```
>> sigma = sigma'

sigma =

         0
    0.0278
         0
         0
         0
         0
```

The strain vector is next obtained by applying (2.2) as follows:

```
>> epsilon = S*sigma

epsilon =

    -0.0000
     0.0023
    -0.0011
```

```
     0
     0
     0
```

Note that the strain in dimensionless. Note also that ε_{11} is very small but is not zero as it seems from the above result. To get the strain ε_{11} exactly, we need to use the `format` command to get more digits as follows:

```
>> format short e
>> epsilon

epsilon =

   -4.4444e-005
    2.2957e-003
   -1.0514e-003
              0
              0
              0
```

Finally, the change in length in each direction is calculated by multiplying the strain by the dimension in each direction as follows:

```
>> d1 = epsilon(1)*60

d1 =

   -2.6667e-003

>> d2 = epsilon(2)*60

d2 =

    1.3774e-001

>> d3 = epsilon(3)*60

d3 =

   -6.3085e-002
```

Notice that the change in the fiber direction is -2.6667×10^{-3} mm which is very small due to the fibers reducing the deformation in this direction. The minus sign indicates that there is a reduction in this dimension along the fibers. The change in the 2-direction is 0.13774 mm and is the largest change because the tensile force is along this direction. This change is positive indicating an extension in the dimension along this direction. Finally, the change in the 3-direction is -0.063085 mm. This change is minus since it indicates a reduction in the dimension along this direction.

Note that you can obtain online help from MATLAB on any of the MATLAB functions by using the `help` command. For example, to obtain help on the MATLAB function *OrthotropicCompliance*, use the `help` command as follows:

```
>> help OrthotropicCompliance

OrthotropicCompliance    This function returns the compliance matrix
                         for orthotropic materials. There are nine
                         arguments representing the nine independent
                         material constants. The size of the compliance
                         matrix is 6 x 6.
```

Note that we can use the MATLAB function *TransverselyIsotropicCompliance* instead of the MATLAB function *OrthotropicCompliance* in this example to obtain the same results. This is because the material constants for graphite-reinforced polymer composite material are the same in the 2- and 3-directions.

MATLAB Example 2.3

Repeat Example 2.2 if the cube is made of aluminum instead of graphite-reinforced polymer composite material. The material constants for aluminum are $E = 72.4\,\text{GPa}$ and $\nu = 0.300$. Use MATLAB.

Solution

This example is solved using MATLAB. First, the normal stress in the 2-direction is calculated in GPa as follows:

```
>> sigma2 = 100/(60*60)

sigma2 =

    0.0278
```

Next, the stress vector is setup directly as a column vector as follows:

```
>> sigma = [0 ; sigma2 ; 0 ; 0 ; 0 ; 0]

sigma =

         0
    0.0278
         0
         0
         0
         0
```

Since aluminum is an isotropic material, the compliance matrix for aluminum is calculated using the MATLAB function *IsotropicCompliance* as follows:

```
>> S = IsotropicCompliance(72.4, 0.3)

S =

    0.0138   -0.0041   -0.0041         0         0         0
   -0.0041    0.0138   -0.0041         0         0         0
   -0.0041   -0.0041    0.0138         0         0         0
         0         0         0    0.0359         0         0
         0         0         0         0    0.0359         0
         0         0         0         0         0    0.0359
```

Next, the strain vector is calculated using (2.2) as follows:

```
>> epsilon = S*sigma

epsilon =

1.0e-003 *

   -0.1151
    0.3837
   -0.1151
         0
         0
         0
```

Finally, the change in length in each direction is calculated by multiplying the strain by the dimension in each direction as follows:

```
>> d1 = epsilon(1)*60

d1 =

  -0.0069

>> d2 = epsilon(2)*60

d2 =

   0.0230

>> d3 = epsilon(3)*60

d3 =

  -0.0069
```

Notice that the change in the 1-direction is -0.0069 mm. The minus sign indicates that there is a reduction in this dimension along 1-direction. The change in the 2-direction is 0.0230 mm and is the largest change because the tensile force is along this direction. This change is positive indicating an extension in the dimension along this direction. Finally, the change in the 3-direction is -0.0069 mm. This change is minus since it indicates a reduction in the dimension along this direction. Also, note that the changes along the 1- and 3-directions are identical since the material is isotropic and these two directions are perpendicular to the 2-direction in which the force is applied.

Problems

Problem 2.1

Derive (2.5) in detail.

Problem 2.2

Discuss the validity of the reciprocity relations given in (2.6).

Problem 2.3

Write the 6×6 compliance matrix for a transversely isotropic material directly in terms of the five independent material constants E_1, E_2, ν_{12}, ν_{23}, and G_{12}.

Problem 2.4

Derive expressions for the elements C_{ij} of the stiffness matrix for a transversely isotropic material directly in terms of the five independent material constants E_1, E_2, ν_{12}, ν_{23}, and G_{12}.

Problem 2.5

Write the 6×6 compliance matrix for an isotropic material directly in terms of the two independent material constants E and ν.

Problem 2.6

Write the 6×6 stiffness matrix for an isotropic material directly in terms of the two independent material constants E and ν.

MATLAB Problem 2.7

Consider a 40-mm cube made of glass-reinforced polymer composite material that is subjected to a compressive force of 150 kN perpendicular to the fiber direction, directed along the 3-direction. The cube is free to expand or contract. Use MATLAB to determine the changes in the 40-mm dimensions of the cube. The material constants for glass-reinforced polymer composite material are given as follows [1]:

$$\begin{aligned} E_1 &= 50.0 \text{ GPa}, & E_2 = E_3 &= 15.20 \text{ GPa} \\ \nu_{23} &= 0.428, & \nu_{12} = \nu_{13} &= 0.254 \\ G_{23} &= 3.28 \text{ GPa}, & G_{12} = G_{13} &= 4.70 \text{ GPa} \end{aligned}$$

MATLAB Problem 2.8

Repeat Problem 2.7 if the cube is made of aluminum instead of glass-reinforced polymer composite material. The material constants for aluminum are $E = 72.4\,\text{GPa}$ and $\nu = 0.300$. Use MATLAB.

MATLAB Problem 2.9

When a fiber-reinforced composite material is heated or cooled, the material expands or contracts just like an isotropic material. This is deformation that takes place independently of any applied load. Let ΔT be the change in temperature and let α_1, α_2, and α_3 be the coefficients of thermal expansion for the composite material in the 1, 2, and 3-directions, respectively. In this case, the stress-strain relation of (2.1) and (2.2) becomes as follows:

$$\begin{Bmatrix} \varepsilon_1 - \alpha_1 \Delta T \\ \varepsilon_2 - \alpha_2 \Delta T \\ \varepsilon_3 - \alpha_3 \Delta T \\ \gamma_{23} \\ \gamma_{13} \\ \gamma_{12} \end{Bmatrix} = \begin{bmatrix} S_{11} & S_{12} & S_{13} & 0 & 0 & 0 \\ S_{12} & S_{22} & S_{23} & 0 & 0 & 0 \\ S_{13} & S_{23} & S_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & S_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & S_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & S_{66} \end{bmatrix} \begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \tau_{23} \\ \tau_{13} \\ \tau_{12} \end{Bmatrix} \tag{2.10}$$

In terms of the stiffness matrix (2.10) becomes as follows:

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \tau_{23} \\ \tau_{13} \\ \tau_{12} \end{Bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & 0 & 0 & 0 \\ C_{12} & C_{22} & C_{23} & 0 & 0 & 0 \\ C_{13} & C_{23} & C_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_1 - \alpha_1 \Delta T \\ \varepsilon_2 - \alpha_2 \Delta T \\ \varepsilon_3 - \alpha_3 \Delta T \\ \gamma_{23} \\ \gamma_{13} \\ \gamma_{12} \end{Bmatrix} \tag{2.11}$$

In (2.10) and (2.11), the strains ε_1, ε_2, and ε_3 are called the *total strains*, $\alpha_1 \Delta T$, $\alpha_2 \Delta T$, and $\alpha_3 \Delta T$ are called the *free thermal strains*, and $(\varepsilon_1 - \alpha_1 \Delta T)$, $(\varepsilon_2 - \alpha_2 \Delta T)$, and $(\varepsilon_3 - \alpha_3 \Delta T)$ are called the *mechanical strains*.

Consider now the cube of graphite-reinforced polymer composite material of Example 2.2 but without the tensile force. Suppose the cube is heated 30°C above some reference state. Given $\alpha_1 = -0.01800 \times 10^{-6}/°\mathrm{C}$ and $\alpha_2 = \alpha_3 = 24.3 \times 10^{-6}/°\mathrm{C}$, use MATLAB to determine the changes in length of the cube in each one of the three directions.

Problem 2.10

Consider the effects of moisture strains in this problem. Let ΔM be the change in moisture and let β_1, β_2, and β_3 be the coefficients of moisture expansion in the 1, 2, and 3-directions, respectively. In this case, the free moisture strains are $\beta_1 \Delta M$, $\beta_2 \Delta M$, and $\beta_3 \Delta M$ in the 1, 2, and 3-directions, respectively. Write the stress-strain equations in this case that correspond to (2.10) and (2.11). In your equations, superimpose both the free thermal strains and the free moisture strains.

3

Elastic Constants Based on Micromechanics

3.1 Basic Equations

The purpose of this chapter is to predict the material constants (also called elastic constants) of a composite material by studying the micromechanics of the problem, i.e. by studying how the matrix and fibers interact. These are the same material constants used in Chap. 2 to calculate the compliance and stiffness matrices. Computing the stresses within the matrix, within the fiber, and at the interface of the matrix and fiber is very important for understanding some of the underlying failure mechanisms. In considering the fibers and surrounding matrix, we have the following assumptions [1]:

1. Both the matrix and fibers are linearly elastic.
2. The fibers are infinitely long.
3. The fibers are spaced periodically in square-packed or hexagonal packed arrays.

There are three different approaches that are used to determine the elastic constants for the composite material based on micromechanics. These three approaches are [1]:

1. Using numerical models such as the finite element method.
2. Using models based on the theory of elasticity.
3. Using rule-of-mixtures models based on a strength-of-materials approach.

Consider a unit cell in either a square-packed array (Fig. 3.1) or a hexagonal-packed array (Fig. 3.2) – see [1]. The ratio of the cross-sectional area of the fiber to the total cross-sectional area of the unit cell is called the *fiber volume fraction* and is denoted by V^f. The fiber volume fraction satisfies the relation $0 < V^f < 1$ and is usually 0.5 or greater. Similarly, the *matrix volume fraction* V^m is the ratio of the cross-sectional area of the matrix to the total cross-sectional area of the unit cell. Note that V^m also satisfies

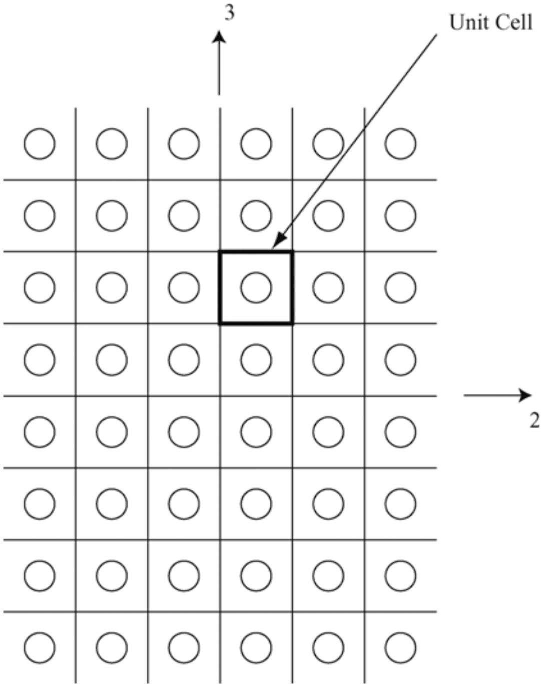

Fig. 3.1. A unit cell in a square-packed array of fiber-reinforced composite material

$0 < V^m < 1$. The following relation can be shown to exist between V^f and V^m:

$$V^f + V^m = 1 \tag{3.1}$$

In the above, we use the notation that a superscript m indicates a matrix quantity while a superscript f indicates a fiber quantity. In addition, the matrix material is assumed to be isotropic so that $E_1^m = E_2^m = E^m$ and $\nu_{12}^m = \nu^m$. However, the fiber material is assumed to be only transversely isotropic such that $E_3^f = E_2^f$, $\nu_{13}^f = \nu_{12}^f$, and $\nu_{23}^f = \nu_{32}^f = \nu^f$.

Using the strength-of-materials approach and the simple rule of mixtures, we have the following relations for the elastic constants of the composite material (see [1]). For Young's modulus in the 1-direction (also called the longitudinal stiffness), we have the following relation:

$$E_1 = E_1^f V^f + E^m V^m \tag{3.2}$$

where E_1^f is Young's modulus of the fiber in the 1-direction while E^m is Young's modulus of the matrix. For Poisson's ratio ν_{12}, we have the following relation:

$$\nu_{12} = \nu_{12}^f V^f + \nu^m V^m \tag{3.3}$$

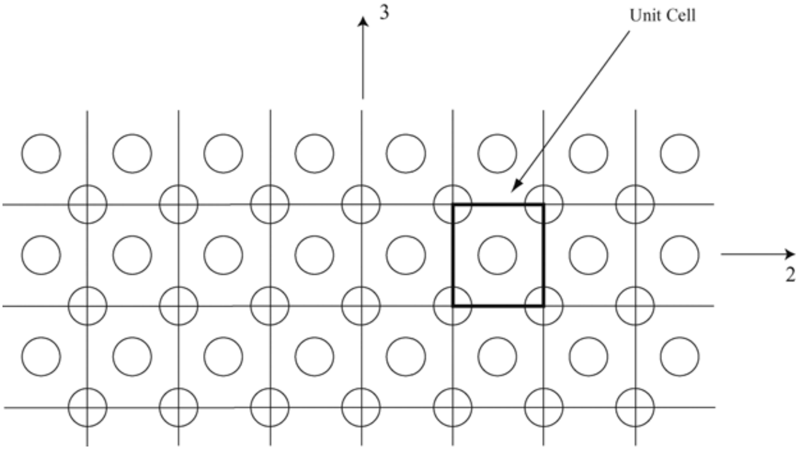

Fig. 3.2. A unit cell in a hexagonal-packed array of fiber-reinforced composite material

where ν_{12}^f and ν^m are Poisson's ratios for the fiber and matrix, respectively. For Young's modulus in the 2-direction (also called the transverse stiffness), we have the following relation:

$$\frac{1}{E_2} = \frac{V^f}{E_2^f} + \frac{V^m}{E^m} \tag{3.4}$$

where E_2^f is Young's modulus of the fiber in the 2-direction while E^m is Young's modulus of the matrix. For the shear modulus G_{12}, we have the following relation:

$$\frac{1}{G_{12}} = \frac{V^f}{G_{12}^f} + \frac{V^m}{G^m} \tag{3.5}$$

where G_{12}^f and G^m are the shear moduli of the fiber and matrix, respectively.

For the coefficients of thermal expansion α_1 and α_2 (see Problem 2.9), we have the following relations:

$$\alpha_1 = \frac{\alpha_1^f E_1^f V^f + \alpha^m E^m V^m}{E_1^f V^f + E^m V^m} \tag{3.6}$$

$$\begin{aligned} \alpha_2 = {} & \left[\alpha_2^f - \left(\frac{E^m}{E_1}\right)\nu_1^f(\alpha^m - \alpha_1^f)V^m\right]V^f \\ & + \left[\alpha^m + \left(\frac{E_1^f}{E_1}\right)\nu^m(\alpha^m - \alpha_1^f)V^f\right]V^m \end{aligned} \tag{3.7}$$

where α_1^f and α_2^f are the coefficients of thermal expansion for the fiber in the 1- and 2-directions, respectively, and α^m is the coefficient of thermal expansion for the matrix. However, we can use a simple rule-of-mixtures relation for α_2 as follows:

$$\alpha_2 = \alpha_2^f V^f + \alpha^m V^m \tag{3.8}$$

A similar simple rule-of-mixtures relation for α_1 cannot be used simply because the matrix and fiber must expand or contract the same amount in the 1-direction when the temperature is changed.

While the simple rule-of-mixtures models used above give accurate results for E_1 and ν_{12}, the results obtained for E_2 and G_{12} do not agree well with finite element analysis and elasticity theory results. Therefore, we need to modify the simple rule-of-mixtures models shown above. For E_2, we have the following modified rule-of-mixtures formula:

$$\frac{1}{E_2} = \frac{\frac{V^f}{E_2^f} + \frac{\eta V^m}{E^m}}{V^f + \eta V^m} \tag{3.9}$$

where η is the stress-partitioning factor (related to the stress σ_2). This factor satisfies the relation $0 < \eta < 1$ and is usually taken between 0.4 and 0.6. Another alternative rule-of-mixtures formula for E_2 is given by:

$$\frac{1}{E_2} = \frac{\eta^f V^f}{E_2^f} + \frac{\eta^m V^m}{E^m} \tag{3.10}$$

where the factors η^f and η^m are given by:

$$\eta^f = \frac{E_1^f V^f + \left[\left(1 - \nu_{12}^f \nu_{21}^f\right) E^m + \nu^m \nu_{21}^f E_1^f\right] V^m}{E_1^f V^f + E^m V^m} \tag{3.11}$$

$$\eta^m = \frac{\left[\left(1 - \nu^{m^2}\right) E_1^f - \left(1 - \nu^m \nu_{12}^f\right) E^m\right] V^f + E^m V^m}{E_1^f V^f + E^m V^m} \tag{3.12}$$

The above alternative model for E_2 gives accurate results and is used whenever the modified rule-of-mixtures model of (3.9) cannot be applied, i.e. when the factor η is not known.

The modified rule-of-mixtures model for G_{12} is given by the following formula:

$$\frac{1}{G_{12}} = \frac{\frac{V^f}{G_{12}^f} + \frac{\eta' V^m}{G^m}}{V^f + \eta' V^m} \tag{3.13}$$

where η' is the shear stress-partitioning factor. Note that η' satisfies the relation $0 < \eta' < 1$ but using $\eta' = 0.6$ gives results that correlate with the elasticity solution.

Finally, the elasticity solution gives the following formula for G_{12}:

$$G_{12} = G^m \left[\frac{(G^m + G_{12}^f) - V^f (G^m - G_{12}^f)}{(G^m + G_{12}^f) + V^f (G^m - G_{12}^f)}\right] \tag{3.14}$$

3.2 MATLAB Functions Used

The six MATLAB functions used in this chapter to calculate the elastic material constants are:

E1 (Vf, E1f, Em) – This function calculates the longitudinal Young's modulus E_1 for the lamina. Its input consists of three arguments as illustrated in the listing below.

NU12 (Vf, NU12f, NUm) – This function calculates Poisson's ratio ν_{12} for the lamina. Its input consists of three arguments as illustrated in the listing below.

E2 (Vf, E2f, Em, Eta, NU12f, NU21f, NUm, E1f, p) – This function calculates the transverse Young's modulus E_2 for the lamina. Its input consists of nine arguments as illustrated in the listing below. Use the value zero for any argument not needed in the calculations.

G12 (Vf, G12f, Gm, EtaPrime, p) – This function calculates the shear modulus G_{12} for the lamina. Its input consists of five arguments as illustrated in the listing below. Use the value zero for any argument not needed in the calculations.

Alpha1 (Vf, E1f, Em, Alpha1f, Alpham) – This function calculates the coefficient of thermal expansion α_1 for the lamina. Its input consists of five arguments as illustrated in the listing below.

Alpha2 (Vf, Alpha2f, Alpham, E1, E1f, Em, NU1f, NUm, Alpha1f, p) – This function calculates the coefficient of thermal expansion α_2 for the lamina. Its input consists of ten arguments as illustrated in the listing below. Use the value zero for any argument not needed in the calculations.

The following is a listing of the MATLAB source code for each function:

```
function y = E1(Vf,E1f,Em)
%E1   This function returns Young's modulus in the
%     longitudinal direction. Its input are three values:
%     Vf  - fiber volume fraction
%     E1f - longitudinal Young's modulus of the fiber
%     Em  - Young's modulus of the matrix
%     This function uses the simple rule-of-mixtures formula
%     of equation (3.2)
Vm = 1 - Vf;
y  = Vf*E1f + Vm*Em;
```

```
function y = NU12(Vf,NU12f,NUm)
%NU12    This function returns Poisson's ratio NU12
%        Its input are three values:
%        Vf    - fiber volume fraction
%        NU12f - Poisson's ratio NU12 of the fiber
%        NUm   - Poisson's ratio of the matrix
```

```
%        This function uses the simple rule-of-mixtures
%        formula of equation (3.3)
Vm = 1 - Vf;
 y = Vf*NU12f + Vm*NUm;
```

```
function y = E2(Vf,E2f,Em,Eta,NU12f,NU21f,NUm,E1f,p)
%E2    This function returns Young's modulus in the
%      transverse direction. Its input are nine values:
%      Vf    - fiber volume fraction
%      E2f   - transverse Young's modulus of the fiber
%      Em    - Young's modulus of the matrix
%      Eta   - stress-partitioning factor
%      NU12f - Poisson's ratio NU12 of the fiber
%      NU21f - Poisson's ratio NU21 of the fiber
%      NUm   - Poisson's ratio of the matrix
%      E1f   - longitudinal Young's modulus of the fiber
%      p     - parameter used to determine which equation to use:
%              p = 1 - use equation (3.4)
%              p = 2 - use equation (3.9)
%              p = 3 - use equation (3.10)
%      Use the value zero for any argument not needed
%      in the calculations.
Vm = 1 - Vf;
if p == 1
    y = 1/(Vf/E2f + Vm/Em);
elseif p == 2
    y = 1/((Vf/E2f + Eta*Vm/Em)/(Vf + Eta*Vm));
elseif p == 3
   deno = E1f*Vf + Em*Vm;
   etaf = (E1f*Vf + ((1-NU12f*NU21f)*Em + NUm*NU21f*E1f)*Vm) /deno;
   etam = (((1-NUm*NUm)*E1f - (1-NUm*NU12f)*Em)*Vf + Em*Vm) /deno;
      y = 1/(etaf*Vf/E2f + etam*Vm/Em);
end
```

```
function y = G12(Vf,G12f,Gm,EtaPrime,p)
%G12   This function returns the shear modulus G12
%      Its input are five values:
%      Vf       - fiber volume fraction
%      G12f     - shear modulus G12 of the fiber
%      Gm       - shear modulus of the matrix
%      EtaPrime - shear stress-partitioning factor
%      p        - parameter used to determine which equation to use:
%                 p = 1 - use equation (3.5)
%                 p = 2 - use equation (3.13)
%                 p = 3 - use equation (3.14)
%      Use the value zero for any argument not needed
%      in the calculations.
Vm = 1 - Vf;
```

```
if p == 1
    y = 1/(Vf/G12f + Vm/Gm);
elseif p == 2
    y = 1/((Vf/G12f + EtaPrime*Vm/Gm)/(Vf + EtaPrime*Vm));
elseif p == 3
    y = Gm*((Gm + G12f) - Vf*(Gm - G12f))/((Gm + G12f) +
        Vf*(Gm - G12f));
end
```

```
function y = Alpha1(Vf,E1f,Em,Alpha1f,Alpham)
%Alpha1   This function returns the coefficient of thermal
%         expansion in the longitudinal direction.
%         Its input are five values:
%         Vf      - fiber volume fraction
%         E1f     - longitudinal Young's modulus of the fiber
%         Em      - Young's modulus of the matrix
%         Alpha1f - coefficient of thermal expansion in the
%                   1-direction for the fiber
%         Alpham  - coefficient of thermal expansion for the matrix
Vm = 1 - Vf;
 y = (Vf*E1f*Alpha1f + Vm*Em*Alpham)/(E1f*Vf + Em*Vm);
```

```
function y = Alpha2(Vf,Alpha2f,Alpham,E1,E1f,Em,NU1f,NUm,
             Alpha1f,p)
%Alpha2   This function returns the coefficient of thermal
%         expansion in the transverse direction.
%         Its input are ten values:
%         Vf     - fiber volume fraction
%         Alpha2f - coefficient of thermal expansion in the
%                  2-direction for the fiber
%         Alpham  - coefficient of thermal expansion for the matrix
%         E1     - longitudinal Young's modulus of the lamina
%         E1f    - longitudianl Young's modulus of the fiber
%         Em     - Young's modulus of the matrix
%         NU1f   - Poisson's ratio of the fiber
%         NUm    - Poisson's ratio of the matrix
%         Alpha1f - coefficient of thermal expansion in the
%                  1-direction
%         p      - parameter used to determine which equation to use
%                  p = 1 - use equation (3.8)
%                  p = 2 - use equation (3.7)
%         Use the value zero for any argument not needed in
%         the calculation
Vm = 1 - Vf;
if p == 1
    y = Vf*Alpha2f + Vm*Alpham;
elseif p == 2
    y = (Alpha2f - (Em/E1)*NU1f*(Alpham - Alpha1f)*Vm)*Vf +
        (Alpham + (E1f/E1)*NUm*(Alpham - Alpha1f)*Vf)*Vm;
end
```

Example 3.1

Derive the simple rule-of-mixtures formula for the calculation of the longitudinal modulus E_1 given in (3.2).

Solution

Consider a longitudinal cross-section of length L of the fiber and matrix in a lamina as shown in Fig. 3.3. Let A^f and A^m be the cross-sectional areas of the fiber and matrix, respectively. Let also F_1^f and F_1^m be the longitudinal forces in the fiber and matrix, respectively. Then we have the following relations:

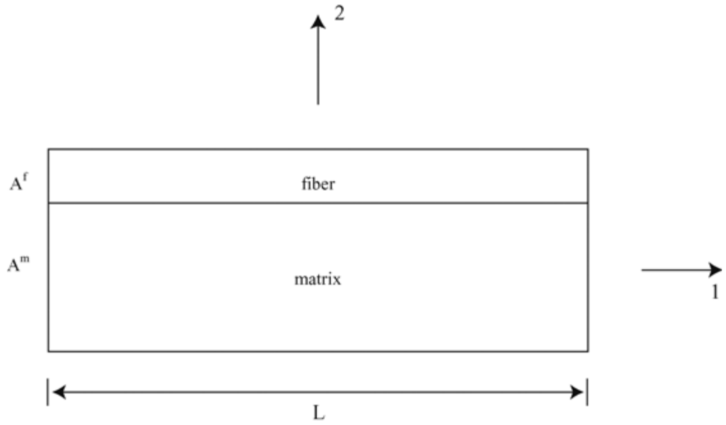

Fig. 3.3. A longitudinal cross-section of fiber-reinforced composite material for Example 3.1

$$F_1^f = \sigma_1^f A^f \tag{3.15a}$$

$$F_1^m = \sigma_1^m A^m \tag{3.15b}$$

where σ_1^f and σ_1^m are the longitudinal normal stresses in the fiber and matrix, respectively. These stresses are given in terms of the longitudinal strains ε_1^f and ε_1^m as follows:

$$\sigma_1^f = E_1^f \varepsilon_1^f \tag{3.16a}$$

$$\sigma_1^m = E^m \varepsilon_1^m \tag{3.16b}$$

where E_1^f is the longitudinal modulus of the fiber and E^m is the modulus of the matrix.

Let F_1 be the total longitudinal force in the lamina where F_1 is given by:

$$F_1 = \sigma_1 A \tag{3.17}$$

where σ_1 is the total longitudinal normal stress in the lamina and A is the total cross-sectional area of the lamina. The total longitudinal normal stress σ_1 is given by:

$$\sigma_1 = E_1 \varepsilon_1 \tag{3.18}$$

However, using force equilibrium, it is clear that we have the following relation between the total longitudinal force and the longitudinal forces in the fiber and matrix:

$$F_1 = F_1^f + F_1^m \tag{3.19}$$

Substituting (3.15a,b) and (3.17) into (3.19), then substituting (3.16a,b) and (3.18) into the resulting equation, we obtain the following relation:

$$E_1 \varepsilon_1 A = E_1^f \varepsilon_1^f A^f + E^m \varepsilon_1^m A^m \tag{3.20}$$

Next, we use the compatibility condition $\varepsilon_1^f = \varepsilon_1^m = \varepsilon_1$ since the matrix, fiber, and lamina all have the same strains. Equation (3.20) is simplified as follows:

$$E_1 A = E_1^f A^f + E^m A^m \tag{3.21}$$

Finally, we divide (3.21) by A and note that $A^f/A = V^f$ and $A^m/A = V^m$ to obtain the required formula for E_1 as follows:

$$E_1 = E_1^f V^f + E^m V^m \tag{3.22}$$

MATLAB Example 3.2

Consider a graphite-reinforced polymer composite lamina with the following material properties for the matrix and fibers [1]:

$$\begin{aligned}
E^m &= 4.62\,\text{GPa}, & \nu^m &= 0.360 \\
E_1^f &= 233\,\text{GPa}, & \nu_{12}^f &= 0.200 \\
E_2^f &= 23.1\,\text{GPa}, & \nu_{23}^f &= 0.400 \\
G_{12}^f &= 8.96\,\text{GPa} & G_{23}^f &= 8.27\,\text{GPa}
\end{aligned}$$

Use MATLAB and the simple rule-of-mixtures formulas to calculate the values of the four elastic constants E_1, ν_{12}, E_2, and G_{12} for the lamina. Use $V^f = 0.6$.

Solution

This example is solved using MATLAB. First, the MATLAB function $E1$ is used to calculate the longitudinal modulus E_1 in GPa as follows:

```
>> E1(0.6, 233, 4.62)

ans =

  141.6480
```

Poisson's ratio ν_{12} is then calculated using the MATLAB function *NU12* as follows:

```
>> NU12(0.6, 0.200, 0.360)

ans =

    0.2640
```

The transverse modulus E_2 is then calculated in GPa using the MATLAB function $E2$ as follows (note that we use the value zero for each parameter not needed in the calculations):

```
>> E2(0.6, 23.1, 4.62, 0, 0, 0, 0, 0, 1)

ans =

    8.8846
```

The shear modulus for the matrix G^m is calculated in GPa using (2.8) as follows:

```
>> Gm = 4.62/(2*(1 + 0.360))

Gm =

    1.6985
```

Finally, the shear modulus G_{12} of the lamina is calculated in GPa using the MATLAB function $G12$ as follows:

```
>> G12(0.6, 8.96, Gm, 0, 1)

ans =

    3.3062
```

Note that ν_{23}^f and G_{23}^f are not used in this example.

MATLAB Example 3.3

Consider the graphite-reinforced polymer composite lamina of Example 3.2. Use MATLAB to plot a graph for each one of the four elastic constants (E_1, ν_{12}, E_2, G_{12}) versus the fiber volume fraction V^f. Use all values of V^f ranging from 0 to 1 (in increments of 0.1).

Solution

This example is solved using MATLAB. First, the array for the x-axis is set up as follows:

```
>> x = [0  0.1  0.2  0.3  0.4  0.5  0.6  0.7  0.8  0.9  1]

x =

  Columns 1 through 10

        0  0.1000  0.2000  0.3000  0.4000  0.5000  0.6000  0.7000
        0.8000  0.9000

  Column 11

        1.0000
```

Then, the longitudinal modulus E_1 is calculated in GPa using the MATLAB function $E1$ for all values of V^f between 0 and 1 as follows (in increments of 0.1):

```
>> y(1) = E1(0,  233,  4.62)

y =

    4.6200

>> y(2) = E1(0.1,  233,  4.62)

y =

    4.6200  27.4580

>> y(3) = E1(0.2,  233,  4.62)

y =

    4.6200  27.4580  50.2960

>> y(4) = E1(0.3,  233,  4.62)
```

```
y =
    4.6200   27.4580   50.2960   73.1340

>> y(5) = E1(0.4,  233,  4.62)

y =

    4.6200   27.4580   50.2960   73.1340   95.9720

>> y(6) = E1(0.5,  233,  4.62)

y =

    4.6200   27.4580   50.2960   73.1340   95.9720   118.8100

>> y(7) = E1(0.6,  233,  4.62)

y =

    4.6200   27.4580   50.2960   73.1340   95.9720   118.8100   141.6480

>> y(8) = E1(0.7,  233,  4.62)

y =

    4.6200   27.4580   50.2960   73.1340   95.9720   118.8100   141.6480
    164.4860

>> y(9) = E1(0.8,  233,  4.62)

y =

    4.6200   27.4580   50.2960   73.1340   95.9720   118.8100   141.6480
    164.4860   187.3240

>> y(10) = E1(0.9,  233,  4.62)

y =

    4.6200   27.4580   50.2960   73.1340   95.9720   118.8100   141.6480
    164.4860   187.3240   210.1620

>> y(11) = E1(1,  233,  4.62)

y =

  Columns 1 through 10

    4.6200   27.4580   50.2960   73.1340   95.9720   118.8100   141.6480
```

```
    164.4860  187.3240  210.1620
  Column 11

    233.0000
```

The plot command is then used to plot the graph of E_1 versus V^f as follows. The resulting plot is shown in Fig. 3.4. Notice that the variation is linear.

```
>> plot(x,y)
>> xlabel('V^f');
>> ylabel('E_1 (GPa)');
```

Poisson's ratio ν_{12} is then calculated using the MATLAB function *NU12* for all values of V^f between 0 and 1 as follows (in increments of 0.1):

```
>> z(1) = NU12(0,  0.200,  0.360)

z =

     0.3600

>> z(2) = NU12(0.1,  0.200,  0.360)

z =

     0.3600  0.3440

>> z(3) = NU12(0.2,  0.200,  0.360)
z =
```

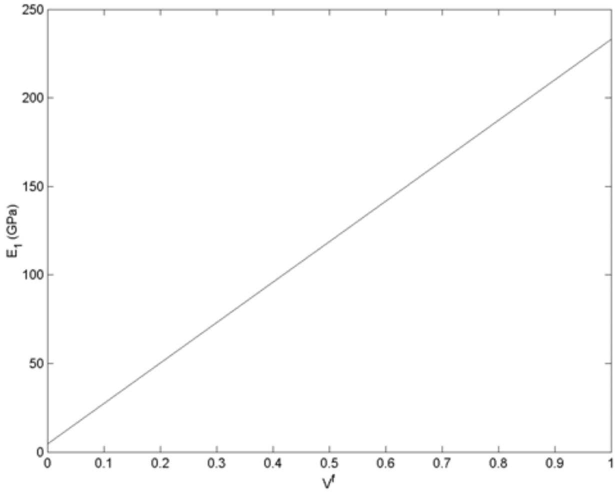

Fig. 3.4. Variation of E_1 versus V^f for Example 3.3

```
     0.3600   0.3440   0.3280

>> z(4) = NU12(0.3,  0.200,  0.360)

z =

     0.3600   0.3440   0.3280   0.3120

>> z(5) = NU12(0.4,  0.200,  0.360)

z =

     0.3600   0.3440   0.3280   0.3120   0.2960

>> z(6) = NU12(0.5,  0.200,  0.360)

z =

     0.3600   0.3440   0.3280   0.3120   0.2960   0.2800

>> z(7) = NU12(0.6,  0.200,  0.360)

z =

     0.3600   0.3440   0.3280   0.3120   0.2960   0.2800   0.2640

>> z(8) = NU12(0.7,  0.200,  0.360)

z =

     0.3600   0.3440   0.3280   0.3120   0.2960   0.2800   0.2640   0.2480


>> z(9) = NU12(0.8,  0.200,  0.360)

z =

     0.3600   0.3440   0.3280   0.3120   0.2960   0.2800   0.2640
     0.2480   0.2320

>> z(10) = NU12(0.9,  0.200,  0.360)

z =

     0.3600   0.3440   0.3280   0.3120   0.2960   0.2800   0.2640   0.2480
     0.2320   0.2160

>> z(11) = NU12(1,  0.200,  0.360)
```

```
z =

  Columns 1 through 10

     0.3600   0.3440   0.3280   0.3120   0.2960   0.2800   0.2640   0.2480
     0.2320   0.2160

  Column 11

       0.2000
```

The plot command is then used to plot the graph of ν_{12} versus V^f as follows. The resulting plot is shown in Fig. 3.5. Notice that the variation is linear.

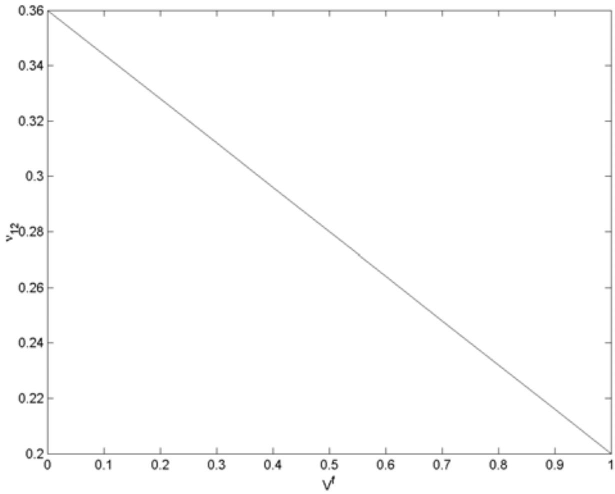

Fig. 3.5. Variation of ν_{12} versus V^f for Example 3.3

```
>> plot(x,z)
>> xlabel('V^f');
>> ylabel('\nu_{12}');
```

The transverse modulus E_2 is then calculated using the MATLAB function $E2$ using all values of V^f between 0 and 1 as follows (in increments of 0.1):

```
>> w(1) = E2(0, 23.1, 4.62, 0, 0, 0, 0, 0, 1)

w =
     4.6200

>> w(2) = E2(0.1,  23.1,  4.62,  0,  0,  0,  0,  0, 1)
```

```
w =
     4.6200  5.0217

>> w(3) = E2(0.2,  23.1,  4.62,  0,  0,  0,  0,  0,  1)

w =
     4.6200  5.0217  5.5000

>> w(4) = E2(0.3,  23.1,  4.62,  0,  0,  0,  0,  0,  1)

w =
     4.6200  5.0217  5.5000  6.0789


>> w(5) = E2(0.4, 23.1, 4.62, 0, 0, 0, 0, 0, 1)

w =
     4.6200  5.0217  5.5000  6.0789  6.7941


>> w(6) = E2(0.5, 23.1, 4.62, 0, 0, 0, 0, 0, 1)

w =
     4.6200 5.0217  5.5000  6.0789  6.7941  7.7000

>> w(7) = E2(0.6,  23.1,  4.62,  0,  0,  0,  0,  0,  1)

w =
     4.6200 5.0217 5.5000 6.0789 6.7941 7.7000 8.8846

>> w(8) = E2(0.7, 23.1, 4.62, 0, 0, 0, 0, 0, 1)

w =
     4.6200 5.0217 5.5000 6.0789 6.7941 7.7000 8.8846 10.5000

>> w(9) = E2(0.8, 23.1, 4.62, 0, 0, 0, 0, 0, 1)

w =
     4.6200  5.0217  5.5000  6.0789  6.7941  7.7000  8.8846  10.5000
     12.8333

>> w(10) = E2(0.9, 23.1, 4.62, 0, 0, 0, 0, 0, 1)

w =
     4.6200  5.0217  5.5000  6.0789  6.7941  7.7000  8.8846  10.5000
     12.8333 16.5000

>> w(11) = E2(1, 23.1, 4.62, 0, 0, 0, 0, 0, 1)
```

```
w =

   Columns 1 through 10
     4.6200   5.0217   5.5000   6.0789   6.7941   7.7000   8.8846   10.5000
     12.8333   16.5000

   Column 11

     23.1000
```

The plot command is then used to plot the graph of E_2 versus V^f as follows. The resulting plot is shown in Fig. 3.6. Notice that the variation is nonlinear.

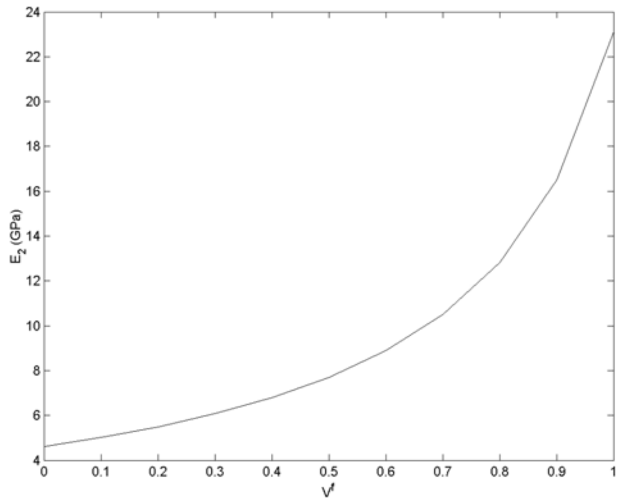

Fig. 3.6. Variation of E_2 versus V^f for Example 3.3

```
>> plot(x,w)
>> xlabel('V^f');
>> ylabel('E_2 (GPa)');
```

Finally, the shear modulus G_{12} is then calculated using the MATLAB function $G12$ using all values of V^f between 0 and 1 as follows (in increments of 0.1). Note that we first calculate G^m using (2.8).

```
>> Gm = 4.62/(2*(1 + 0.360))


Gm =
     1.6985


>> u(1) = G12(0, 8.96, Gm, 0, 1)
```

```
u =
     1.6985

>> u(2) = G12(0.1, 8.96, Gm, 0, 1)

u =
     1.6985   1.8483

>> u(3) = G12(0.2, 8.96, Gm, 0, 1)

u =
     1.6985 1.8483 2.0271

>> u(4) = G12(0.3, 8.96, Gm, 0, 1)

u =
     1.6985 1.8483 2.0271 2.2441

>> u(5) = G12(0.4,  8.96,  Gm,  0,  1)

u =
     1.6985  1.8483  2.0271  2.2441  2.5133

>> u(6) = G12(0.5,  8.96,  Gm,  0,  1)

u =
     1.6985  1.8483  2.0271  2.2441  2.5133  2.8557

>> u(7) = G12(0.6, 8.96, Gm, 0, 1)
u =
     1.6985 1.8483 2.0271 2.2441 2.5133 2.8557 3.3062

>> u(8) = G12(0.7,  8.96,  Gm,  0,  1)

u =
     1.6985  1.8483  2.0271  2.2441  2.5133  2.8557  3.3062  3.9254

>> u(9) = G12(0.8,  8.96,  Gm,  0,  1)

u =
     1.6985  1.8483  2.0271  2.2441  2.5133  2.8557  3.3062  3.9254
     4.8301

>> u(10) = G12(0.9, 8.96, Gm, 0, 1)

u =
     1.6985  1.8483  2.0271  2.2441  2.5133  2.8557  3.3062  3.9254
     4.8301  6.2766
```

```
>> u(11) = G12(1, 8.96, Gm, 0, 1)

u =

  Columns 1 through 10

    1.6985  1.8483  2.0271  2.2441  2.5133  2.8557  3.3062  3.9254
    4.8301  6.2766

  Column 11

    8.9600
```

The plot command is then used to plot the graph of G_{12} versus V^f as follows. The resulting plot is shown in Fig. 3.7. Notice that the variation is nonlinear.

```
>> plot(x,u)
>> xlabel('V^f');
>> ylabel('G_{12} (GPa)');
```

Problems

Problem 3.1

Derive (3.1) in detail.

Problem 3.2

Derive the simple rule-of-mixtures formula for the calculation of Poisson's ratio ν_{12} given in (3.3).

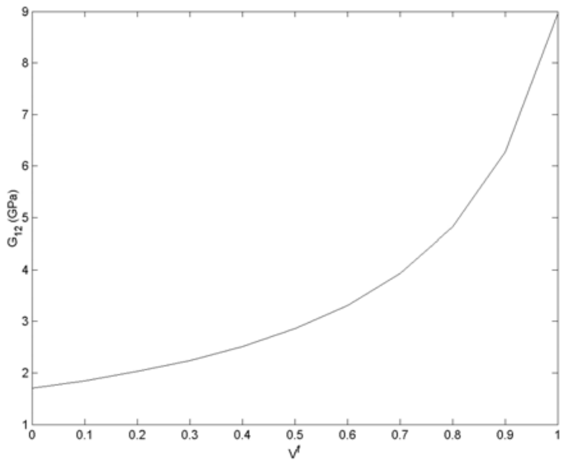

Fig. 3.7. Variation of G_{12} versus V^f for Example 3.3

Problem 3.3

Derive the simple rule-of-mixtures formula for the calculation of the transverse modulus E_2 given in (3.4).

MATLAB Problem 3.4

In the calculation of the transverse modulus E_2 using the simple rule-of-mixtures formula of (3.4), the results can be improved by replacing E^m by $E^{m'}$ where $E^{m'}$ is given by:

$$E^{m'} = \frac{E^m}{1 - \nu^{m^2}} \tag{3.23}$$

where ν^m is Poisson's ratio of the matrix. Modify the MATLAB function E_2 with the addition of this formula as a fourth case to be calculated when the parameter p is set to the value 4.

MATLAB Problem 3.5

Consider a carbon/epoxy composite lamina with the following matrix and fiber material properties [2]:

$$E_2^f = 14.8 \text{ GPa}, \quad E^m = 3.45 \text{ GPa}, \quad \nu^m = 0.36$$

Use MATLAB to calculate the transverse modulus E_2 using the following three methods (use $V^f = 0.65$):

(a) the simple rule-of-mixtures formula of (3.4).
(b) the modified rule-of-mixtures formula of (3.9) with $\eta = 0.5$.
(c) the alternative rule-of-mixtures formula of (3.10). For this case, use $E_1^f =$ 85.6 GPa, $\nu_{12}^f = \nu_{21}^f = 0.3$.

MATLAB Problem 3.6

Consider the glass/epoxy composite lamina of Problem 3.5. Use MATLAB to plot a graph of the transverse modulus E_2 versus the fiber volume fraction V^f for each one of the following cases. Use all values of V^f ranging from 0 to 1 (in increments of 0.1).

(a) the simple rule-of-mixtures formula of (3.4).
(b) the modified rule-of-mixtures formula of (3.9) with $\eta = 0.4$.
(c) the modified rule-of-mixtures formula of (3.9) with $\eta = 0.5$.
(d) the modified rule-of-mixtures formula of (3.9) with $\eta = 0.6$.
(e) the alternative rule-of-mixtures formula of (3.10) with the values given in part (c) of Problem 3.5.

Make sure that all five graphs appear on the same plot.

MATLAB Problem 3.7

Consider a carbon/epoxy composite lamina with the following matrix and fiber material properties [2]:

$$G_{12}^f = 28.3 \text{ GPa}, \quad G^m = 1.27 \text{ GPa}$$

Use MATLAB to calculate the shear modulus G_{12} using the following three methods (use $V^f = 0.55$):

(a) the simple rule-of-mixtures formula of (3.5).
(b) the modified rule-of-mixtures formula of (3.13) with $\eta' = 0.6$.
(c) the elasticity formula of (3.14).

MATLAB Problem 3.8

Consider the glass/epoxy composite lamina of Problem 3.7. Use MATLAB to plot a graph of the shear modulus G_{12} versus the fiber volume fraction V^f for each one of the following cases. Use all values of V^f ranging from 0 to 1 (in increments of 0.1).

(a) the simple rule-of-mixtures formula of (3.5).
(b) the modified rule-of-mixtures formula of (3.13) with $\eta' = 0.6$.
(c) the elasticity formula of (3.14).

Make sure that all three graphs appear on the same plot.

MATLAB Problem 3.9

Consider the graphite-reinforced polymer composite lamina of Example 3.2. Let the coefficients of thermal expansion for the matrix and fibers be given as follows [1]:

$$\begin{aligned} \alpha^m &= 41.4 \times 10^{-6}/\text{K} \\ \alpha_1^f &= -0.540 \times 10^{-6}/\text{K} \\ \alpha_2^f &= 10.10 \times 10^{-6}/\text{K} \end{aligned}$$

Use MATLAB to calculate α_1 and α_2 for the lamina. When calculating α_2, use the two formulas given (3.7) and (3.8).

Problem 3.10

Consider a fiber-reinforced composite lamina assuming the existence of an interface region. Let E^f, E^m, and E^i be Young's moduli for the matrix, fiber, and interface material, respectively. Also, let V^f, V^m, and V^i be the volume fractions of the fiber, matrix, and interface satisfying the relation $V^f + V^m + V^i = 1$. Determine an expression for the longitudinal modulus E_1 of the lamina using a simple rule-of-mixtures formula.

4

Plane Stress

4.1 Basic Equations

In the analysis of fiber-reinforced composite materials, the assumption of *plane stress* is usually used for each layer (lamina). This is mainly because fiber-reinforced materials are utilized in beams, plates, cylinders, and other structural shapes which have at least one characteristic geometric dimension in an order of magnitude less than the other two dimensions. In this case, the stress components σ_3, τ_{23}, and τ_{13} are set to zero with the assumption that the 1-2 plane of the principal material coordinate system is in the plane of the layer (lamina) – see [1]. Therefore, the stresses σ_1, σ_2, and τ_{12} lie in a plane, while the stresses σ_3, τ_{23}, and τ_{13} are perpendicular to this plane and are zero (see Fig. 4.1).

Using the assumption of plane stress, it is seen that the stress-strain relations of Chap. 2 are greatly simplified. Setting $\sigma_3 = \tau_{23} = \tau_{13} = 0$ in (2.1) leads to the following:

$$\begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \gamma_{23} \\ \gamma_{13} \\ \gamma_{12} \end{Bmatrix} = \begin{bmatrix} S_{11} & S_{12} & S_{13} & 0 & 0 & 0 \\ S_{12} & S_{22} & S_{23} & 0 & 0 & 0 \\ S_{13} & S_{23} & S_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & S_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & S_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & S_{66} \end{bmatrix} \begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ 0 \\ 0 \\ 0 \\ \tau_{12} \end{Bmatrix} \tag{4.1}$$

As a result of the plane stress assumption and using (4.1), we conclude that:

$$\gamma_{23} = 0 \tag{4.2}$$

$$\gamma_{13} = 0 \tag{4.3}$$

$$\varepsilon_3 = S_{13}\sigma_1 + S_{23}\sigma_2 \neq 0 \tag{4.4}$$

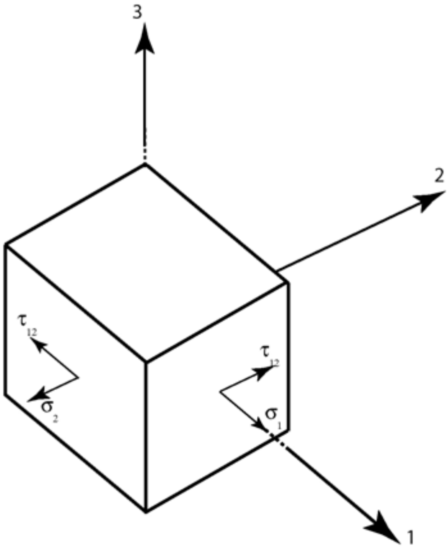

Fig. 4.1. An infinitesimal fiber-reinforced composite element in a state of plane stress

Therefore (4.1) reduces to the following equation:

$$\begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \gamma_{12} \end{Bmatrix} = \begin{bmatrix} S_{11} & S_{12} & 0 \\ S_{12} & S_{22} & 0 \\ 0 & 0 & S_{66} \end{bmatrix} \begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{Bmatrix} \tag{4.5}$$

The 3×3 matrix in (4.5) is called the *reduced compliance matrix*. The inverse of the reduced compliance matrix is the *reduced stiffness matrix* given as follows:

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{Bmatrix} = \begin{bmatrix} Q_{11} & Q_{12} & 0 \\ Q_{12} & Q_{22} & 0 \\ 0 & 0 & Q_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \gamma_{12} \end{Bmatrix} \tag{4.6}$$

where the elements Q_{ij} are given as follows:

$$Q_{11} = \frac{S_{22}}{S_{11}S_{22} - S_{12}^2} \tag{4.7a}$$

$$Q_{12} = -\frac{S_{12}}{S_{11}S_{22} - S_{12}^2} \tag{4.7b}$$

$$Q_{22} = \frac{S_{11}}{S_{11}S_{22} - S_{12}^2} \tag{4.7c}$$

$$Q_{66} = \frac{1}{S_{66}} \tag{4.7d}$$

4.2 MATLAB Functions Used

The four MATLAB functions used in this chapter to calculate the reduced compliance and stiffness matrices are:

ReducedCompliance(E1, E2, NU12, G12) – This function calculates the reduced compliance matrix for the lamina. Its input consists of four arguments representing the four elastic constants E_1, E_2, ν_{12}, and G_{12}. See Problem 4.1.

ReducedStiffness(E1, E2, NU12, G12) – This function calculates the reduced stiffness matrix for the lamina. Its input consists of four arguments representing the four elastic constants E_1, E_2, ν_{12}, and G_{12}. See Problem 4.2.

ReducedIsotropicCompliance(E, NU) – This function calculates the reduced isotropic compliance matrix for the lamina. Its input consists of two arguments representing the two elastic constants E and ν. See Problem 4.3.

ReducedIsotropicStiffness(E, NU) – This function calculates the reduced isotropic stiffness matrix for the lamina. Its input consists of two arguments representing the two elastic constants E and ν. See Problem 4.4.

The following is a listing of the MATLAB source code for each function:

```
function y = ReducedCompliance(E1,E2,NU12,G12)
%ReducedCompliance   This function returns the reduced compliance
%                    matrix for fiber-reinforced materials.
%                    There are four arguments representing four
%                    material constants. The size of the reduced
%                    compliance matrix is 3 x 3.
y = [1/E1 -NU12/E1 0 ; -NU12/E1 1/E2 0 ; 0 0 1/G12];
```

```
function y = ReducedStiffness(E1,E2,NU12,G12)
%ReducedStiffness   This function returns the reduced stiffness
%                   matrix for fiber-reinforced materials.
%                   There are four arguments representing four
%                   material constants. The size of the reduced
%                   stiffness matrix is 3 x 3.
NU21 = NU12*E2/E1;
y = [E1/(1-NU12*NU21) NU12*E2/(1-NU12*NU21) 0 ;NU12*E2/(1-NU12*NU21)
    E2/(1-NU12*NU21) 0 ; 0 0 G12];
```

```
function y = ReducedIsotropicCompliance(E,NU)
%ReducedIsotropicCompliance   This function returns the
%                             reduced isotropic compliance
%                             matrix for fiber-reinforced materials.
%                             There are two arguments representing
%                             two material constants. The size of
%                             the reduced compliance matrix is 3 x 3.
y = [1/E -NU/E 0 ; -NU/E 1/E 0 ; 0 0 2*(1+NU)/E];
```

```
function y = ReducedIsotropicStiffness(E,NU)
%ReducedIsotropicStiffness   This function returns the
%                            reduced isotropic stiffness
%                            matrix for fiber-reinforced materials.
%                            There are two arguments representing
%                            two material constants. The size of
%                            the reduced stiffness matrix is 3 x 3.
y = [E/(1-NU*NU) NU*E/(1-NU*NU) 0 ; NU*E/(1-NU*NU) E/(1-NU*NU) 0 ; 0
    0   E/2/(1+NU)];
```

Example 4.1

Derive the following expressions for the elements Q_{ij} of the 3×3 reduced stiffness matrix where C_{ij} are the elements of the 6×6 stiffness matrix of (2.3).

$$Q_{11} = C_{11} - \frac{C_{13}^2}{C_{33}} \tag{4.8a}$$

$$Q_{12} = C_{12} - \frac{C_{13}C_{23}}{C_{33}} \tag{4.8b}$$

$$Q_{22} = C_{22} - \frac{C_{23}^2}{C_{33}} \tag{4.8c}$$

$$Q_{66} = C_{66} \tag{4.8d}$$

Solution

For the case of plane stress, set $\sigma_3 = \tau_{23} = \tau_{13} = 0$ in (2.3) to obtain (while using the symmetric form of the $[C]$ matrix).

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ 0 \\ 0 \\ 0 \\ \tau_{12} \end{Bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & 0 & 0 & 0 \\ C_{12} & C_{22} & C_{23} & 0 & 0 & 0 \\ C_{13} & C_{23} & C_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \gamma_{23} \\ \gamma_{13} \\ \gamma_{12} \end{Bmatrix} \tag{4.9}$$

We can therefore write the following three equations based on the first, second, and sixth rows of (4.9):

$$\sigma_1 = C_{11}\varepsilon_1 + C_{12}\varepsilon_2 + C_{13}\varepsilon_3 \tag{4.10a}$$

$$\sigma_2 = C_{12}\varepsilon_1 + C_{22}\varepsilon_2 + C_{23}\varepsilon_3 \tag{4.10b}$$

$$\tau_{12} = C_{66}\gamma_{12} \tag{4.10c}$$

In addition, we can write the following relation based on the third row of (4.9):

$$0 = C_{13}\varepsilon_1 + C_{23}\varepsilon_2 + C_{33}\varepsilon_3 \tag{4.11}$$

Solving (4.11) for ε_3 to obtain:

$$\varepsilon_3 = -\frac{C_{13}}{C_{33}}\varepsilon_1 - \frac{C_{23}}{C_{33}}\varepsilon_2 \tag{4.12}$$

Substitute (4.12) into (4.10a,b) and simplify to obtain the following relations:

$$\sigma_1 = \left(C_{11} - \frac{C_{13}^2}{C_{33}}\right)\varepsilon_1 + \left(C_{12} - \frac{C_{13}C_{23}}{C_{33}}\right)\varepsilon_2 \tag{4.13a}$$

$$\sigma_2 = \left(C_{12} - \frac{C_{13}C_{23}}{C_{33}}\right)\varepsilon_1 + \left(C_{22} - \frac{C_{23}^2}{C_{33}}\right)\varepsilon_2 \tag{4.13b}$$

$$\tau_{12} = C_{66}\gamma_{12} \tag{4.13c}$$

Rewriting (4.13a,b,c) in matrix form we obtain (see (4.6)):

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{Bmatrix} = \begin{bmatrix} Q_{11} & Q_{12} & 0 \\ Q_{12} & Q_{22} & 0 \\ 0 & 0 & Q_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \gamma_{12} \end{Bmatrix} \tag{4.14}$$

where the elements Q_{ij} are given by (see (4.8a,b,c,d).):

$$Q_{11} = C_{11} - \frac{C_{13}^2}{C_{33}} \tag{4.15a}$$

$$Q_{12} = C_{12} - \frac{C_{13}C_{23}}{C_{33}} \tag{4.15b}$$

$$Q_{22} = C_{22} - \frac{C_{23}^2}{C_{33}} \tag{4.15c}$$

$$Q_{66} = C_{66} \tag{4.15d}$$

MATLAB Example 4.2

Consider a layer of graphite-reinforced composite material 200 mm long, 100 mm wide, and 0.200 mm thick. The layer is subjected to an inplane tensile force of 4 kN in the fiber direction which is perpendicular to the 100-mm width. Assume the layer to be in a state of plane stress and use the elastic constants given in Example 2.2. Use MATLAB to determine the transverse strain ε_3.

Solution

This example is solved using MATLAB. First, the full 6×6 compliance matrix is obtained as follows using the MATLAB function
OrthotropicCompliance of Chap. 2.

```
>> S = OrthotropicCompliance(155.0, 12.10, 12.10, 0.248, 0.458,
       0.248, 4.40, 3.20, 4.40)

S =

    0.0065   -0.0016   -0.0016         0         0         0
   -0.0016    0.0826   -0.0379         0         0         0
   -0.0016   -0.0379    0.0826         0         0         0
         0         0         0    0.3125         0         0
         0         0         0         0    0.2273         0
         0         0         0         0         0    0.2273
```

Using the third row of (4.1), we obtain the following expression for the transverse strain ε_3 (see (4.4)):

$$\varepsilon_3 = S_{13}\sigma_1 + S_{23}\sigma_2 \tag{4.16}$$

Next, the stresses σ_1 and σ_2 are calculated in GPa as follows:

```
>> sigma1 = 4/(100*0.200)

sigma1 =

    0.2000

>> sigma2 = 0

sigma2 =

    0
```

Finally, the transverse strain ε_3 is calculated using (4.16) as follows:

```
>> epsilon3 = S(1,3)*sigma1 + S(2,3)*sigma2

epsilon3 =

   -3.2000e-004
```

Thus, we obtain the transverse strain $\varepsilon_3 = -3.2 \times 10^{-4}$.

MATLAB Example 4.3

Consider the graphite-reinforced composite material of Example 2.2.

(a) Use MATLAB to determine the reduced compliance and stiffness matrices.
(b) Use MATLAB to check that the two matrices obtained in (a) are indeed inverses of each other by multiplying them together to get the identity matrix.

Solution

This example is solved using MATLAB. First, the reduced compliance matrix is obtained as follows using the MATLAB function *ReducedCompliance*.

```
>> S = ReducedCompliance(155.0, 12.10, 0.248, 4.40)

S =

    0.0065   -0.0016         0
   -0.0016    0.0826         0
         0         0    0.2273
```

Next, the reduced stiffness matrix is obtained as follows using the MATLAB function *ReducedStiffness*:

```
>> Q = ReducedStiffness(155.0, 12.10, 0.248, 4.40)

Q =

  155.7478    3.0153         0
    3.0153   12.1584         0
         0         0    4.4000
```

Finally, the two matrices are multiplied with each other to get the identity matrix in order to show that they are indeed inverses of each other.

```
>> S*Q

ans =

    1.0000         0         0
   -0.0000    1.0000         0
         0         0    1.0000
```

Problems

Problem 4.1

Write the reduced compliance matrix for a fiber-reinforced composite material in terms of the four elastic constants E_1, E_2, ν_{12}, and G_{12}.

Problem 4.2

Write the reduced stiffness matrix for a fiber-reinforced composite material in terms of the four elastic constants E_1, E_2, ν_{12}, and G_{12}.

Problem 4.3

Write the reduced compliance matrix for an isotropic fiber-reinforced composite material in terms of the two elastic constants E and ν.

Problem 4.4

Write the reduced stiffness matrix for an isotropic fiber-reinforced composite material in terms of the two elastic constants E and ν.

MATLAB Problem 4.5

Consider the glass-reinforced polymer composite material of Problem 2.7.

(a) Use MATLAB to determine the reduced compliance and stiffness matrices.
(b) Use MATLAB to check that the two matrices obtained in (a) are indeed inverses of each other by multiplying them together to get the identity matrix.

MATLAB Problem 4.6

Consider the layer of composite material of Example 4.2. Suppose that the layer is subjected to an inplane compressive force of 2.5 kN in the 2-direction instead of the 4 kN force in the 1-direction. Use MATLAB to calculate the transverse strain ε_3 in this case.

MATLAB Problem 4.7

Consider the isotropic material aluminum with $E = 72.4$ GPa and $\nu = 0.3$.

(a) Use MATLAB to determine the reduced compliance and stiffness matrices.
(b) Use MATLAB to check that the two matrices obtained in (a) are indeed inverses of each other by multiplying them together to get the identity matrix.

MATLAB Problem 4.8

Suppose in Example 4.2 that the fibers are perpendicular to the 200-mm direction. Use MATLAB to calculate the transverse strain ε_3 in this case.

MATLAB Problem 4.9

Write two MATLAB functions called *ReducedStiffness2* and *ReducedIsotropicStifness2* where the reduced stiffness matrix in each case is determined by taking the inverse of the reduced compliance matrix.

Problem 4.10

Consider a layer of fiber-reinforced composite material that is subjected to both temperature and moisture variations. Write the 3×3 reduced stress-strain equations that correspond to (4.5) and (4.6). See Problems 2.9 and 2.10 of Chap. 2.

5

Global Coordinate System

5.1 Basic Equations

In this chapter, we will refer the response of each layer (lamina) of material to the same global system. We accomplish this by transforming the stress-strain relations for the lamina 1-2-3 coordinate system into the *global coordinate system*. This transformation will be done for the state of plane stress using the standard transformation relations for stresses and strains given in introductory courses in mechanics of materials [1].

Consider an isolated infinitesimal element in the principal material coordinate system (1-2-3 system) that will be transformed into the x-y-z global coordinate system as shown in Fig. 5.1. The fibers are oriented at angle θ with respect to the $+x$ axis of the global system. The fibers are parallel to the x-y plane and the 3 and z axes coincide. The orientation angle θ will be considered positive when the fibers rotate counterclockwise from the $+x$ axis toward the $+y$ axis.

The stresses on the small volume of element are now identified with respect to the x-y-z system. The six components of stress are now σ_x, σ_y, σ_z, τ_{yz}, τ_{xz}, and τ_{xy}, while the six components of strain are ε_x, ε_y, ε_z, γ_{yz}, γ_{xz}, and γ_{xy} (see Fig. 5.2).

Note that in a plane stress state, it follows that the out-of-plane stress components in the x-y-z global coordinate system are zero, i.e. $\sigma_z = \tau_{yz} = \tau_{xz} = 0$ (see Problem 5.1).

The stress transformation relation is given as follows for the case of plane stress:

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{Bmatrix} = \begin{bmatrix} m^2 & n^2 & 2mn \\ n^2 & m^2 & -2mn \\ -mn & mn & m^2 - n^2 \end{bmatrix} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} \tag{5.1}$$

where $m = \cos\theta$ and $n = \sin\theta$. The above relation is written in compact form as follows:

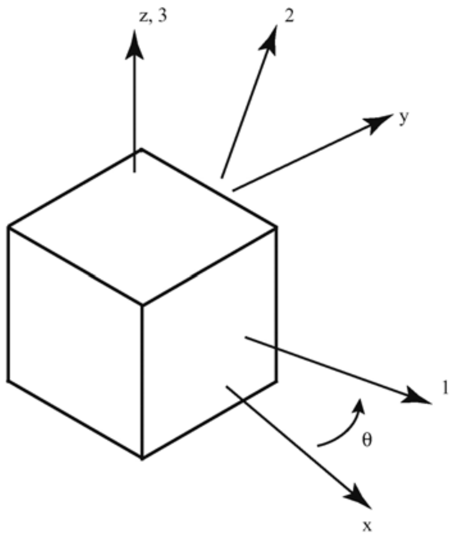

Fig. 5.1. A infinitesimal fiber-reinforced composite element showing the local and global coordinate systems

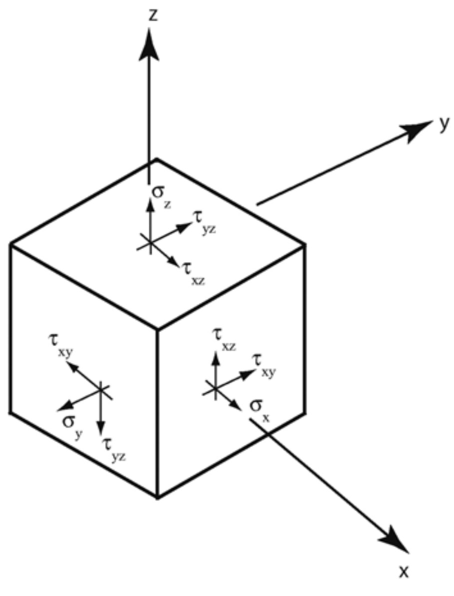

Fig. 5.2. An infinitesimal fiber-reinforced composite element showing the stress components in the global coordinate system

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{Bmatrix} = [T] \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} \tag{5.2}$$

where $[T]$ is the transformation matrix given as follows:

$$[T] = \begin{bmatrix} m^2 & n^2 & 2mn \\ n^2 & m^2 & -2mn \\ -mn & mn & m^2 - n^2 \end{bmatrix} \tag{5.3}$$

The inverse of the matrix $[T]$ is $[T]^{-1}$ given as follows (see Problem 5.3):

$$[T]^{-1} = \begin{bmatrix} m^2 & n^2 & -2mn \\ n^2 & m^2 & 2mn \\ mn & -mn & m^2 - n^2 \end{bmatrix} \tag{5.4}$$

where $[T]^{-1}$ is used in the following equation:

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = [T]^{-1} \begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{Bmatrix} \tag{5.5}$$

Similar transformation relations hold for the strains as follows:

$$\begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \frac{1}{2}\gamma_{12} \end{Bmatrix} = [T] \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \frac{1}{2}\gamma_{xy} \end{Bmatrix} \tag{5.6}$$

$$\begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \frac{1}{2}\gamma_{xy} \end{Bmatrix} = [T]^{-1} \begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \frac{1}{2}\gamma_{12} \end{Bmatrix} \tag{5.7}$$

Note that the strain transformation (5.6) and (5.7) include a factor of 1/2 with the engineering shear strain. Therefore (4.5) and (4.6) of Chap. 4 are modified now to include this factor as follows:

$$\begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \frac{1}{2}\gamma_{12} \end{Bmatrix} = \begin{bmatrix} S_{11} & S_{12} & 0 \\ S_{12} & S_{22} & 0 \\ 0 & 0 & \frac{1}{2}S_{66} \end{bmatrix} \begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{Bmatrix} \tag{5.8}$$

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{Bmatrix} = \begin{bmatrix} Q_{11} & Q_{12} & 0 \\ Q_{12} & Q_{22} & 0 \\ 0 & 0 & 2Q_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \frac{1}{2}\gamma_{12} \end{Bmatrix} \tag{5.9}$$

Substitute (5.6) and (5.2) into (5.8) and rearrange the terms to obtain (also multiply the third row through by a factor of 2):

$$\begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{bmatrix} \bar{S}_{11} & \bar{S}_{12} & \bar{S}_{16} \\ \bar{S}_{12} & \bar{S}_{22} & \bar{S}_{26} \\ \bar{S}_{16} & \bar{S}_{26} & \bar{S}_{66} \end{bmatrix} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} \tag{5.10}$$

where the *transformed reduced compliance matrix* $[\bar{S}]$ is given by:

$$[\bar{S}] = \begin{bmatrix} \bar{S}_{11} & \bar{S}_{12} & \bar{S}_{16} \\ \bar{S}_{12} & \bar{S}_{22} & \bar{S}_{26} \\ \bar{S}_{16} & \bar{S}_{26} & \bar{S}_{66} \end{bmatrix} = [T]^{-1} \begin{bmatrix} S_{11} & S_{12} & 0 \\ S_{12} & S_{22} & 0 \\ 0 & 0 & S_{66} \end{bmatrix} [T] \tag{5.11}$$

Equation (5.11) represents the complex relations that describe the response of an element of fiber-reinforced composite material in a state of plane stress that is subjected to stresses not aligned with the fibers, nor perpendicular to the fibers. In this case, normal stresses cause shear strains and shear stresses cause extensional strains. This coupling found in fiber-reinforced composite materials is called *shear-extension coupling.*

Similarly, we can derive the *transformed reduced stiffness matrix* $[\bar{Q}]$ by substituting (5.2) and (5.6) into (5.9) and rearranging the terms. We therefore obtain:

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \begin{bmatrix} \bar{Q}_{11} & \bar{Q}_{12} & \bar{Q}_{16} \\ \bar{Q}_{12} & \bar{Q}_{22} & \bar{Q}_{26} \\ \bar{Q}_{16} & \bar{Q}_{26} & \bar{Q}_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} \tag{5.12}$$

where $[\bar{Q}]$ is given by:

$$[\bar{Q}] = \begin{bmatrix} \bar{Q}_{11} & \bar{Q}_{12} & \bar{Q}_{16} \\ \bar{Q}_{12} & \bar{Q}_{22} & \bar{Q}_{26} \\ \bar{Q}_{16} & \bar{Q}_{26} & \bar{Q}_{66} \end{bmatrix} = [T]^{-1} \begin{bmatrix} Q_{11} & Q_{12} & 0 \\ Q_{12} & Q_{22} & 0 \\ 0 & 0 & Q_{66} \end{bmatrix} [T] \tag{5.13}$$

Equation (5.13) further supports the shear-extension coupling of fiber-reinforced composite materials. Note that the following relations hold between $[\bar{S}]$ and $[\bar{Q}]$:

$$[\bar{Q}] = [\bar{S}]^{-1} \tag{5.14a}$$

$$[\bar{S}] = [\bar{Q}]^{-1} \tag{5.14b}$$

5.2 MATLAB Functions Used

The four MATLAB functions used in this chapter to calculate the four major matrices are:

T(theta) – This function calculates the transformation matrix $[T]$ given the angle "theta". The orientation angle "theta" must be given in degrees. The returned matrix has size 3×3.

Tinv(theta) – This function calculates the inverse of the transformation matrix $[T]$ given the angle "theta". The orientation angle "theta" must be given in degrees. The returned matrix has size 3×3.

Sbar(S,theta) – This function calculates the transformed reduced compliance matrix $[\bar{S}]$ for the lamina. Its input consists of two arguments representing the reduced compliance matrix $[S]$ and the orientation angle "theta". The returned matrix has size 3×3.

Qbar(Q,theta) – This function calculates the transformed reduced stiffness matrix $[\bar{Q}]$ for the lamina. Its input consists of two arguments representing the reduced stiffness matrix $[Q]$ and the orientation angle "theta". The returned matrix has size 3×3.

The following is a listing of the MATLAB source code for each function:

```
function y = T(theta)
%T   This function returns the transformation matrix T
%    given the orientation angle "theta".
%    There is only one argument representing "theta"
%    The size of the matrix is 3 x 3.
%    The angle "theta" must be given in degrees.
m = cos(theta*pi/180);
n = sin(theta*pi/180);
y = [m*m n*n 2*m*n ; n*n m*m -2*m*n ; -m*n m*n m*m-n*n];
```

```
function y = Tinv(theta)
%Tinv   This function returns the inverse of the
%       transformation matrix T
%       given the orientation angle "theta".
%       There is only one argument representing "theta"
%       The size of the matrix is 3 x 3.
%       The angle "theta" must be given in degrees.
m = cos(theta*pi/180);
n = sin(theta*pi/180);
y = [m*m n*n -2*m*n ; n*n m*m 2*m*n ; m*n -m*n m*m-n*n];
```

```
function y = Sbar(S,theta)
%Sbar   This function returns the transformed reduced
%       compliance matrix "Sbar" given the reduced
%       compliance matrix S and the orientation
%       angle "theta".
%       There are two arguments representing S and "theta"
%       The size of the matrix is 3 x 3.
%       The angle "theta" must be given in degrees.
m = cos(theta*pi/180);
```

```
n = sin(theta*pi/180);
T = [m*m n*n 2*m*n ; n*n m*m -2*m*n ; -m*n m*n m*m-n*n];
Tinv = [m*m n*n -2*m*n ; n*n m*m 2*m*n ; m*n -m*n m*m-n*n];
y = Tinv*S*T;
```

```
function y = Qbar(Q,theta)
%Qbar   This function returns the transformed reduced
%       stiffness matrix "Qbar" given the reduced
%       stiffness matrix Q and the orientation
%       angle "theta".
%       There are two arguments representing Q and "theta"
%       The size of the matrix is 3 x 3.
%       The angle "theta" must be given in degrees.
m = cos(theta*pi/180);
n = sin(theta*pi/180);
T = [m*m n*n 2*m*n ; n*n m*m -2*m*n ; -m*n m*n m*m-n*n];
Tinv = [m*m n*n -2*m*n ; n*n m*m 2*m*n ; m*n -m*n m*m-n*n];
y = Tinv*Q*T;
```

Example 5.1

Using (5.11), derive explicit expressions for the elements $\bar{S}_{ij}$ in terms of S_{ij} and θ (use m and n for θ).

Solution

Multiply the three matrices in (5.11) as follows:

$$\begin{bmatrix} \bar{S}_{11} & \bar{S}_{12} & \bar{S}_{16} \\ \bar{S}_{12} & \bar{S}_{22} & \bar{S}_{26} \\ \bar{S}_{16} & \bar{S}_{26} & \bar{S}_{66} \end{bmatrix} = \begin{bmatrix} m^2 & n^2 & -2mn \\ n^2 & m^2 & 2mn \\ mn & -mn & m^2-n^2 \end{bmatrix} \begin{bmatrix} S_{11} & S_{12} & 0 \\ S_{12} & S_{22} & 0 \\ 0 & 0 & S_{66} \end{bmatrix} \begin{bmatrix} m^2 & n^2 & 2mn \\ n^2 & m^2 & -2mn \\ -mn & mn & m^2-n^2 \end{bmatrix} \tag{5.15}$$

The above multiplication can be performed either manually or using a computer algebra system like MAPLE or MATHEMATICA or the MATLAB Symbolic Math Toolbox. Therefore, we obtain the following expression:

$$\bar{S}_{11} = S_{11}m^4 + (2S_{12} + S_{66})n^2m^2 + S_{22}n^4 \tag{5.16a}$$

$$\bar{S}_{12} = (S_{11} + S_{22} - S_{66})n^2m^2 + S_{12}(n^4 + m^4) \tag{5.16b}$$

$$\bar{S}_{16} = (2S_{11} - 2S_{12} - S_{66})nm^3 - (2S_{22} - 2S_{12} - S_{66})n^3m \tag{5.16c}$$

$$\bar{S}_{22} = S_{11}n^4 + (2S_{12} + S_{66})n^2m^2 + S_{22}m^4 \tag{5.16d}$$

$$\bar{S}_{26} = (2S_{11} - 2S_{12} - S_{66})n^3m - (2S_{22} - 2S_{12} - S_{66})nm^3 \tag{5.16e}$$

$$\bar{S}_{66} = 2(2S_{11} + 2S_{22} - 4S_{12} - S_{66})n^2m^2 + S_{66}(n^4 + m^4) \tag{5.16f}$$

MATLAB Example 5.2

Consider a graphite-reinforced polymer composite lamina with the elastic constants as given in Example 2.2. Use MATLAB to plot the values of the six elements $\bar{S}_{ij}$ of the transformed reduced compliance matrix $[\bar{S}]$ as a function of the orientation angle θ in the range $-\pi/2 \leq \theta \leq \pi/2$.

Solution

This example is solved using MATLAB. First, the reduced 3×3 compliance matrix is obtained as follows using the MATLAB function *ReducedCompliance* of Chap. 4.

```
>> S = ReducedCompliance(155.0, 12.10, 0.248, 4.40)

S =

    0.0065   -0.0016         0
   -0.0016    0.0826         0
         0         0    0.2273
```

Next, the transformed reduced compliance matrix $[\bar{S}]$ is calculated at each value of θ between $-90°$ and $90°$ in increments of $10°$ using the MATLAB function *Sbar*.

```
>> S1 = Sbar(S, -90)

S1 =

    0.0826   -0.0016   -0.0000
   -0.0016    0.0065    0.0000
   -0.0000    0.0000    0.2273

>> S2 = Sbar(S, -80)

S2 =

    0.0909   -0.0122   -0.0452
   -0.0122    0.0193    0.0712
   -0.0226    0.0356    0.2061

>> S3 = Sbar(S, -70)

S3 =

    0.1111   -0.0390   -0.0647
   -0.0390    0.0528    0.1137
   -0.0323    0.0568    0.1524
```

```
>> S4 = Sbar(S, -60)

S4 =

    0.1315   -0.0695   -0.0454
   -0.0695    0.0934    0.1114
   -0.0227    0.0557    0.0914

>> S5 = Sbar(S, -50)

S5 =

    0.1390   -0.0894    0.0065
   -0.0894    0.1258    0.0685
    0.0033    0.0342    0.0516

>> S6 = Sbar(S, -40)

S6 =

    0.1258   -0.0894    0.0685
   -0.0894    0.1390    0.0065
    0.0342    0.0033    0.0516

>> S7 = Sbar(S, -30)

S7 =

    0.0934   -0.0695    0.1114
   -0.0695    0.1315   -0.0454
    0.0557   -0.0227    0.0914


>> S8 = Sbar(S, -20)

S8 =

    0.0528   -0.0390    0.1137
   -0.0390    0.1111   -0.0647
    0.0568   -0.0323    0.1524

>> S9 = Sbar(S, -10)

S9 =

    0.0193   -0.0122    0.0712
   -0.0122    0.0909   -0.0452
    0.0356   -0.0226    0.2061
```

```
>> S10 = Sbar(S, 0)

S10 =

    0.0065   -0.0016         0
   -0.0016    0.0826         0
         0         0    0.2273

>> S11 = Sbar(S, 10)

S11 =

    0.0193   -0.0122   -0.0712
   -0.0122    0.0909    0.0452
   -0.0356    0.0226    0.2061

>> S12 = Sbar(S, 20)

S12 =

    0.0528   -0.0390   -0.1137
   -0.0390    0.1111    0.0647
   -0.0568    0.0323    0.1524

>> S13 = Sbar(S, 30)

S13 =

    0.0934   -0.0695   -0.1114
   -0.0695    0.1315    0.0454
   -0.0557    0.0227    0.0914

>> S14 = Sbar(S, 40)

S14 =

    0.1258   -0.0894   -0.0685
   -0.0894    0.1390   -0.0065
   -0.0342   -0.0033    0.0516

>> S15 = Sbar(S, 50)

S15 =

    0.1390   -0.0894   -0.0065
   -0.0894    0.1258   -0.0685
   -0.0033   -0.0342    0.0516

>> S16 = Sbar(S, 60)
```

```
S16 =

    0.1315   -0.0695    0.0454
   -0.0695    0.0934   -0.1114
    0.0227   -0.0557    0.0914

>> S17 = Sbar(S, 70)

S17 =

    0.1111    -0.0390   0.0647
   -0.0390     0.0528  -0.1137
    0.0323    -0.0568   0.1524

>> S18 = Sbar(S, 80)

S18 =

    0.0909    -0.0122   0.0452
   -0.0122     0.0193  -0.0712
    0.0226    -0.0356   0.2061

>> S19 = Sbar(S, 90)

S19 =

    0.0826    -0.0016   0.0000
   -0.0016     0.0065  -0.0000
    0.0000    -0.0000   0.2273
```

The x-axis is now setup for the plots as follows:

```
>> x = [-90  -80  -70  -60  -50  -40  -30  -20  -10  0  10  20  30 40
       50  60  70  80  90]

x =

   -90  -80  -70  -60  -50  -40  -30  -20  -10  0  10  20  30  40  50
   60  70   80   90
```

The values of $\bar{S}_{11}$ are now calculated for each value of θ between -90° and 90° in increments of 10°.

```
>> y1 = [S1(1,1) S2(1,1) S3(1,1) S4(1,1) S5(1,1) S6(1,1) S7(1,1)
        S8(1,1) S9(1,1) S10(1,1) S11(1,1) S12(1,1) S13(1,1) S14(1,1)
        S15(1,1) 16(1,1) S17(1,1) S18(1,1) S19(1,1)]

y1 =

   Columns 1 through 14
```

```
    0.0826    0.0909    0.1111    0.1315    0.1390    0.1258    0.0934
    0.0528    0.0193    0.0065    0.0193    0.0528    0.0934    0.1258

  Columns 15 through 19

    0.1390    0.1315    0.1111    0.0909    0.0826
```

The plot of the values of $\bar{S}_{11}$ versus θ is now generated using the following commands and is shown in Fig. 5.3. Notice that this compliance is an even function of θ. Notice also the rapid variation of the compliance as θ increases or decreases from $0°$.

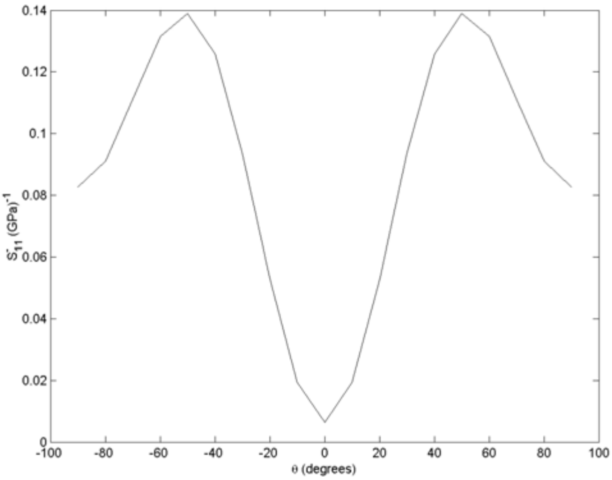

Fig. 5.3. Variation of $\bar{S}_{11}$ versus θ for Example 5.2

```
>> plot(x,y1)
>> xlabel('\theta (degrees)');
>> ylabel('S^{-}_{11} GPa');
```

The values of $\bar{S}_{12}$ are now calculated for each value of θ between $-90°$ and $90°$ in increments of $10°$.

```
>> y2 = [S1(1,2)  S2(1,2)  S3(1,2)  S4(1,2)  S5(1,2)  S6(1,2) S7(1,2)
      S8(1,2) S9(1,2)  S10(1,2)  S11(1,2)  S12(1,2) S13(1,2)
      S14(1,2)  S15(1,2)  S16(1,2) S17(1,2)  S18(1,2) S19(1,2)]

y2 =

  Columns 1 through 14

   -0.0016   -0.0122   -0.0390   -0.0695   -0.0894   -0.0894   -0.0695
   -0.0390   -0.0122   -0.0016   -0.0122   -0.0390   -0.0695   -0.0894
```

```
  Columns 15 through 19

    -0.0894   -0.0695   -0.0390   -0.0122   -0.0016
```

The plot of the values of $\bar{S}_{12}$ versus θ is now generated using the following commands and is shown in Fig. 5.4. Notice that this compliance is an even function of θ. Notice also the rapid variation of the compliance as θ increases or decreases from $0°$.

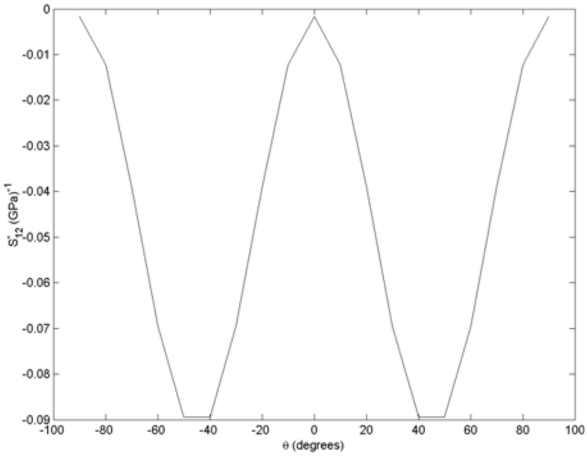

Fig. 5.4. Variation of $\bar{S}_{12}$ versus θ for Example 5.2

```
>> plot(x,y2)
>> xlabel('\theta (degrees)');
>> ylabel('S^{-}_{12} GPa');
```

The values of $\bar{S}_{16}$ are now calculated for each value of θ between $-90°$ and $90°$ in increments of $10°$.

```
>> y3 = [S1(1,3) S2(1,3) S3(1,3) S4(1,3) S5(1,3) S6(1,3) S7(1,3)
        S8(1,3) S9(1,3) S10(1,3) S11(1,3) S12(1,3) S13(1,3) S14(1,3)
        S15(1,3) S16(1,3) S17(1,3) S18(1,3) S19(1,3)]

y3 =

  Columns 1 through 14

    -0.0000   -0.0452   -0.0647   -0.0454    0.0065    0.0685    0.1114
    0.1137    0.0712          0   -0.0712   -0.1137   -0.1114   -0.0685

 Columns 15 through 19

    -0.0065    0.0454    0.0647    0.0452    0.0000
```

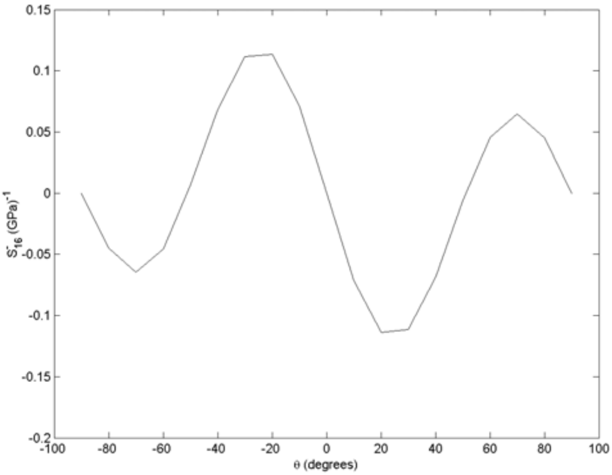

Fig. 5.5. Variation of $\bar{S}_{16}$ versus θ for Example 5.2

The plot of the values of $\bar{S}_{16}$ versus θ is now generated using the following commands and is shown in Fig. 5.5. Notice that this compliance is an odd function of θ. Notice also the rapid variation of the compliance as θ increases or decreases from $0°$.

```
>> plot(x,y3)
>> xlabel('\theta (degrees)');
>> ylabel('S^{-}_{16} GPa');
```

The values of $\bar{S}_{22}$ are now calculated for each value of θ between $-90°$ and $90°$ in increments of $10°$.

```
>> y4 = [S1(2,2) S2(2,2) S3(2,2) S4(2,2) S5(2,2) S6(2,2) S7(2,2)
        S8(2,2) S9(2,2) S10(2,2) S11(2,2) S12(2,2) S13(2,2) S14(2,2)
        S15(2,2) S16(2,2) S17(2,2) S18(2,2) S19(2,2)]

y4 =

  Columns 1 through 14

    0.0065    0.0193    0.0528    0.0934    0.1258    0.1390 0.1315
    0.1111    0.0909    0.0826    0.0909    0.1111    0.1315    0.1390

  Columns 15 through 19

    0.1258    0.0934    0.0528    0.0193    0.0065
```

The plot of the values of $\bar{S}_{22}$ versus θ is now generated using the following commands and is shown in Fig. 5.6. Notice that this compliance is an even function of θ. Notice also the rapid variation of the compliance as θ increases or decreases from $0°$.

```
>> plot(x,y4)}
>> xlabel('\theta (degrees)');
>> ylabel('S^{-}_{22} GPa');
```

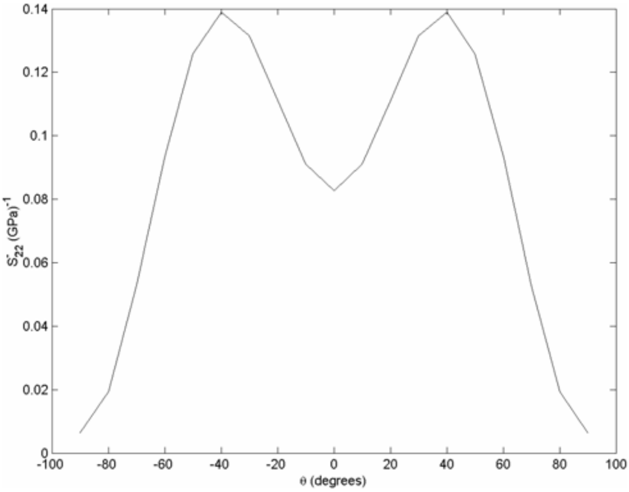

Fig. 5.6. Variation of $\bar{S}_{22}$ versus θ for Example 5.2

The values of $\bar{S}_{26}$ are now calculated for each value of θ between $-90°$ and $90°$ in increments of $10°$

```
>> y5 = [S1(2,3) S2(2,3) S3(2,3) S4(2,3) S5(2,3) S6(2,3) S7(2,3)
       S8(2,3) S9(2,3) S10(2,3) S11(2,3) S12(2,3) S13(2,3) S14(2,3)
       S15(2,3) S16(2,3) S17(2,3) S18(2,3) S19(2,3)]

y5 =

  Columns 1 through 14

    0.0000    0.0712    0.1137    0.1114    0.0685    0.0065   -0.0454
   -0.0647   -0.0452         0    0.0452    0.0647    0.0454   -0.0065

  Columns 15 through 19

   -0.0685   -0.1114   -0.1137   -0.0712   -0.0000
```

The plot of the values of $\bar{S}_{26}$ versus θ is now generated using the following commands and is shown in Fig. 5.7. Notice that this compliance is an odd function of θ. Notice also the rapid variation of the compliance as θ increases or decreases from $0°$.

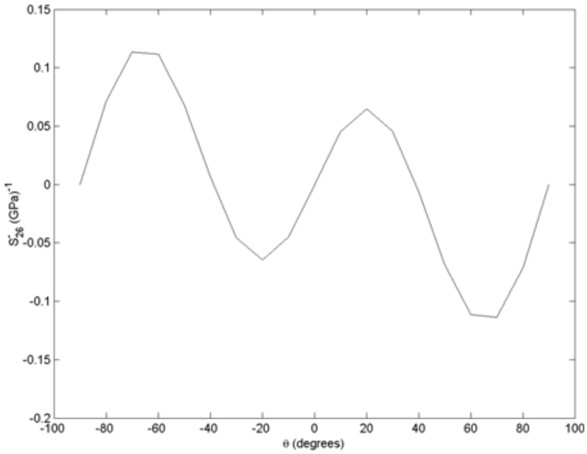

Fig. 5.7. Variation of $\bar{S}_{26}$ versus θ for Example 5.2

```
>> plot(x,y5)
>> xlabel('\theta (degrees)');
>> ylabel('S^{-}_{26} GPa');}
```

The values of $\bar{S}_{66}$ are now calculated for each value of θ between $-90°$ and $90°$ in increments of $10°$.

```
>> y6 = [S1(3,3) S2(3,3) S3(3,3) S4(3,3) S5(3,3) S6(3,3) S7(3,3)
        S8(3,3) S9(3,3) S10(3,3) S11(3,3) S12(3,3) S13(3,3) S14(3,3)
        S15(3,3) S16(3,3) S17(3,3) S18(3,3) S19(3,3)]

y6 =

  Columns 1 through 14

    0.2273    0.2061    0.1524    0.0914    0.0516    0.0516   0.0914
    0.1524    0.2061    0.2273    0.2061    0.1524    0.0914    0.0516

  Columns 15 through 19

    0.0516    0.0914    0.1524    0.2061    0.2273
```

The plot of the values of $\bar{S}_{66}$ versus θ is now generated using the following commands and is shown in Fig. 5.8. Notice that this compliance is an even function of θ. Notice also the rapid variation of the compliance as θ increases or decreases from $0°$.

```
>> plot(x,y6)
>> xlabel('\theta (degrees)');
>> ylabel('S^{-}_{66} GPa');
```

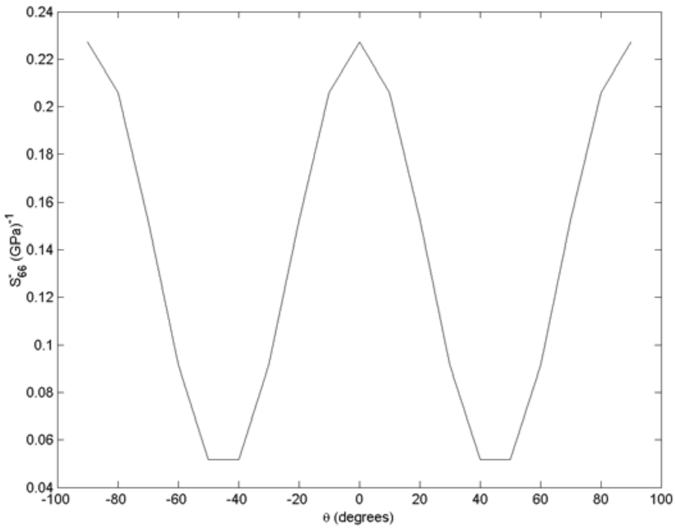

Fig. 5.8. Variation of $\bar{S}_{66}$ versus θ for Example 5.2

MATLAB Example 5.3

Consider a plane element of size 40 mm × 40 mm made of graphite-reinforced polymer composite material whose elastic constants are given in Example 2.2. The element is subjected to a tensile stress $\sigma_x = 200$ MPa in the x-direction. Use MATLAB to calculate the strains and the deformed dimensions of the element in the following two cases:

(a) the fibers are aligned along the x-axis.
(b) the fibers are inclined to the x-axis with an orientation angle $\theta = 30°$.

Solution

This example is solved using MATLAB. First, the reduced compliance matrix is obtained as follows using the MATLAB function *ReducedCompliance* of Chap. 4.

```
>> S = ReducedCompliance(155.0, 12.10, 0.248, 4.40)

S =

    0.0065   -0.0016         0
   -0.0016    0.0826         0
         0         0    0.2273
```

Next, the transformed reduced compliance matrix is calculated for part (a) with $\theta = 0°$ using the MATLAB function *Sbar*.

```
>> S1 = Sbar(S,0)
```

```
S1 =

    0.0065   -0.0016         0
   -0.0016    0.0826         0
         0         0    0.2273
```

Next, the stress vector in the global coordinate system is setup in GPa as follows:

```
>> sigma = [200e-3 ; 0 ; 0]

sigma =

    0.2000
         0
         0
```

The strain vector is now calculated in the global coordinate system using (5.10):

```
>> epsilon = S1*sigma

epsilon =

    0.0013
   -0.0003
         0
```

The change in the length in both the x- and y-direction is calculated next in mm as follows:

```
>> deltax = 40*epsilon(1)

deltax =

    0.0516

>> deltay = 40*epsilon(2)

deltay =

   -0.0128
```

The change in the right angle (in radians) of the element is then calculated using the shear strain obtained from the strain vector above. It is noticed that in this case, this change is zero indicating that the right angle remains a right angle after deformation. This is mainly due to the fibers being aligned along the x-direction.

```
>> gammaxy = epsilon(3)

gammaxy =

     0
```

The deformed dimensions are next calculated as follows:

```
>> dx = 40 + deltax

dx =

   40.0516

>> dy = 40 + deltay

dy =

   39.9872
```

Next, the transformed reduced compliance matrix is calculated for part (b) with $\theta = 30^\circ$ using the MATLAB function *Sbar*.

```
>> S2 = Sbar(S, 30)

S2 =

    0.0934   -0.0695   -0.1114
   -0.0695    0.1315    0.0454
   -0.0557    0.0227    0.0914
```

The strain vector is now calculated in the global coordinate system using (5.10):

```
>> epsilon = S2*sigma

epsilon =

    0.0187
   -0.0139
   -0.0111
```

The change in the length in both the x- and y-direction is calculated next in mm as follows:

```
>> deltax = 40*epsilon(1)

deltax =

    0.7474

>> deltay = 40*epsilon(2)

deltay =

   -0.5562
```

The deformed dimensions are next calculated as follows:

```
>> dx = 40 + deltax
```

```
dx =

   40.7474

>> dy = 40 + deltay

dy =

   39.4438
```

The change in the right angle (in radians) of the element is then calculated using the shear strain obtained from the strain vector above. It is noticed that in this case, there is a negative shear strain indicating that the right angle increases to become more than 90° after deformation. This is mainly due to the fibers being inclined at an angle to the x-direction.

```
>> gammaxy = epsilon(3)

gammaxy =

   -0.0111
```

Problems

Problem 5.1

Show mathematically why the three stresses σ_z, τ_{yz}, and τ_{xz} (with respect to the global coordinate system) vanish in the case of plane stress.

Problem 5.2

Derive (5.1) in detail.

Problem 5.3

Derive the expression for $[T]^{-1}$ given in (5.4). Use (5.3) in your derivation.

Problem 5.4

Show the validity of (5.14a,b).

Problem 5.5

Using (5.13), derive explicit expressions for the elements $\bar{Q}_{ij}$ in terms of Q_{ij} and θ (use m and n for θ).

MATLAB Problem 5.6

Write a new MATLAB function called *Tinv2* which calculates the inverse of the transformation matrix $[T]$ by calculating first $[T]$ then inverting it using the MATLAB function `inv`. Use the same argument "theta" that was used in the MATLAB function *Tinv*.

MATLAB Problem 5.7

(a) Write a new MATLAB function called *Sbar2* to calculate the transformed reduced compliance matrix $[\bar{S}]$. Use the two arguments S and T instead of S and "theta" as was used in the MATLAB function *Sbar*.
(b) Write a new MATLAB function called *Qbar2* to calculate the transformed reduced stiffness matrix $[\bar{Q}]$. Use the two arguments Q and T instead of Q and "theta" as was used in the MATLAB function *Qbar*.

MATLAB Problem 5.8

Consider a glass-reinforced polymer composite lamina with the elastic constants as given in Problem 2.7. Use MATLAB to plot the values of the six elements $\bar{S}_{ij}$ of the transformed reduced compliance matrix $[\bar{S}]$ as a function of the orientation angle θ in the range $-\pi/2 \leq \theta \leq \pi/2$.

MATLAB Problem 5.9

Consider a graphite-reinforced polymer composite lamina with the elastic constants as given in Example 2.2. Use MATLAB to plot the values of the six elements $\bar{Q}_{ij}$ of the transformed reduced stiffness matrix $[\bar{Q}]$ as a function of the orientation angle θ in the range $-\pi/2 \leq \theta \leq \pi/2$.

MATLAB Problem 5.10

Consider a glass-reinforced polymer composite lamina with the elastic constants as given in Problem 2.7. Use MATLAB to plot the values of the six elements $\bar{Q}_{ij}$ of the transformed reduced stiffness matrix $[\bar{Q}]$ as a function of the orientation angle θ in the range $-\pi/2 \leq \theta \leq \pi/2$.

Problem 5.11

(a) Show that the transformed reduced compliance matrix $[\bar{S}]$ becomes equal to the reduced compliance matrix $[S]$ when $\theta = 0°$.
(b) Show that the transformed reduced stiffness matrix $[\bar{Q}]$ becomes equal to the reduced stiffness matrix $[Q]$ when $\theta = 0°$.

Problem 5.12

Show that $[\bar{S}] = [S]$ for isotropic materials. In particular, show the following relation:

$$[\bar{S}] = [S] = \begin{bmatrix} \dfrac{1}{E} & -\dfrac{\nu}{E} & 0 \\ -\dfrac{\nu}{E} & \dfrac{1}{E} & 0 \\ 0 & 0 & \dfrac{2(1+\nu)}{E} \end{bmatrix} \tag{5.17}$$

Problem 5.13

Show that $[\bar{Q}] = [Q]$ for isotropic materials. In particular, show the following relation:

$$[\bar{Q}] = [Q] = \begin{bmatrix} \dfrac{E}{1-\nu^2} & \dfrac{\nu E}{1-\nu^2} & 0 \\ \dfrac{\nu E}{1-\nu^2} & \dfrac{E}{1-\nu^2} & 0 \\ 0 & 0 & \dfrac{E}{2(1+\nu)} \end{bmatrix} \tag{5.18}$$

MATLAB Problem 5.14

Consider a plane element of size 50 mm × 50 mm made of glass-reinforced polymer composite material whose elastic constants are given in Problem 2.7. The element is subjected to a tensile stress $\sigma_x = 100\,\text{MPa}$ in the x-direction. Use MATLAB to calculate the strains and the deformed dimensions of the element in the following three cases:

(a) the fibers are aligned along the x-axis.
(b) the fibers are inclined to the x-axis with an orientation angle $\theta = 45°$.
(c) the fibers are inclined to the x-axis with an orientation angle $\theta = -45°$.

Problem 5.15

Consider the case of free thermal and moisture strains. Show that in this case (5.10) and (5.12) take the following modified forms:

$$\begin{Bmatrix} \varepsilon_x - \alpha_x \Delta T - \beta_x \Delta M \\ \varepsilon_y - \alpha_y \Delta T - \beta_y \Delta M \\ \gamma_{xy} - \alpha_{xy} \Delta T - \beta_{xy} \Delta M \end{Bmatrix} = \begin{bmatrix} \bar{S}_{11} & \bar{S}_{12} & \bar{S}_{16} \\ \bar{S}_{12} & \bar{S}_{22} & \bar{S}_{26} \\ \bar{S}_{16} & \bar{S}_{26} & \bar{S}_{66} \end{bmatrix} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} \tag{5.19}$$

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \begin{bmatrix} \bar{Q}_{11} & \bar{Q}_{12} & \bar{Q}_{16} \\ \bar{Q}_{12} & \bar{Q}_{22} & \bar{Q}_{26} \\ \bar{Q}_{16} & \bar{Q}_{26} & \bar{Q}_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_x - \alpha_x \Delta T - \beta_x \Delta M \\ \varepsilon_y - \alpha_y \Delta T - \beta_y \Delta M \\ \gamma_{xy} - \alpha_{xy} \Delta T - \beta_{xy} \Delta M \end{Bmatrix} \tag{5.20}$$

where ΔT and ΔM are the changes in temperature and moisture, respectively, α_x, α_y and α_{xy} are the coefficients of thermal expansion with respect to the global coordinate system, and β_x, β_y, and β_{xy} are the coefficients of moisture deformation with respect to the global coordinate system.

6

Elastic Constants Based on Global Coordinate System

6.1 Basic Equations

The engineering properties or elastic constants were introduced in Chap. 2 with respect to the lamina 1-2-3 coordinate system. Their evaluation was presented in Chap. 3 based also on the 1-2-3 coordinate system. We can also define elastic constants with respect to the x-y-z global coordinate system. The elastic constants in the x-y-z coordinate system can be derived directly from their definitions, just as they were derived in Chap. 3 for the 1-2-3 coordinate system.

The elastic constants based on the x-y-z global coordinate system are given as follows [1]:

$$E_x = \frac{E_1}{m^4 + \left(\frac{E_1}{G_{12}} - 2\nu_{12}\right) n^2 m^2 + \frac{E_1}{E_2} n^4} \tag{6.1}$$

$$\nu_{xy} = \frac{\nu_{12}\left(n^4 + m^4\right) - \left(1 + \frac{E_1}{E_2} - \frac{E_1}{G_{12}}\right) n^2 m^2}{m^4 + \left(\frac{E_1}{G_{12}} - 2\nu_{12}\right) n^2 m^2 + \frac{E_1}{E_2} n^2} \tag{6.2}$$

$$E_y = \frac{E_2}{m^4 + \left(\frac{E_2}{G_{12}} - 2\nu_{21}\right) n^2 m^2 + \frac{E_2}{E_1} n^4} \tag{6.3}$$

$$\nu_{yx} = \frac{\nu_{21}\left(n^4 + m^4\right) - \left(1 + \frac{E_2}{E_1} - \frac{E_2}{G_{12}}\right) n^2 m^2}{m^4 + \left(\frac{E_2}{G_{12}} - 2\nu_{21}\right) n^2 m^2 + \frac{E_2}{E_1} n^2} \tag{6.4}$$

$$G_{xy} = \frac{G_{12}}{n^4 + m^4 + 2\left(\frac{2G_{12}}{E_1}\left(1 + 2\nu_{12}\right) + \frac{2G_{12}}{E_2} - 1\right) n^2 m^2} \tag{6.5}$$

It is useful to define several other material properties for fiber-reinforced composite materials that can be used to categorize response [1]. These properties have as their basis the fact that an element of fiber-reinforced composite material with its fiber oriented at some arbitrary angle exhibits a shear strain when subjected to a normal stress, and it also exhibits an extensional strain when subjected to a shear stress.

Poisson's ratio is defined as the ratio of extensional strains, given that the element is subjected only to a simple normal stress. By analogy, the *coefficient of mutual influence of the second kind* is defined as the ratio of a shear strain to an extensional strain, given that the element is subjected to only a single normal stress. The *coefficient of mutual influence of the first kind* is defined as the ratio of an extensional strain to a shear strain, given that the element is subjected to only a single shear stress (see [1]).

One coefficient of mutual influence of the second kind is defines as follows:

$$\eta_{xy,x} = \frac{\gamma_{xy}}{\varepsilon_x} \tag{6.6}$$

where $\sigma_x \neq 0$ and all other stresses are zero. Another coefficient of mutual influence of the second kind is defined as follows:

$$\eta_{xy,y} = \frac{\gamma_{xy}}{\varepsilon_y} \tag{6.7}$$

where $\sigma_y \neq 0$ and all other stresses are zero. It can be shown that the coefficients of mutual influence of the second kind can be written as follows:

$$\eta_{xy,x} = \frac{\bar{S}_{16}}{\bar{S}_{11}} \tag{6.8}$$

$$\eta_{xy,y} = \frac{\bar{S}_{26}}{\bar{S}_{22}} \tag{6.9}$$

The coefficients of mutual influence of the first kind are defined as follows:

$$\eta_{x,xy} = \frac{\varepsilon_x}{\gamma_{xy}} \tag{6.10}$$

$$\eta_{y,xy} = \frac{\varepsilon_y}{\gamma_{xy}} \tag{6.11}$$

where $\tau_{xy} \neq 0$ and all other stresses are zero. It can be shown that the coefficients of mutual influence of the first kind can be written as follows:

$$\eta_{x,xy} = \frac{\bar{S}_{16}}{\bar{S}_{66}} \tag{6.12}$$

$$\eta_{y,xy} = \frac{\bar{S}_{26}}{\bar{S}_{66}} \tag{6.13}$$

6.2 MATLAB Functions Used

The nine MATLAB functions used in this chapter to calculate the constants based on the global coordinate system are :

Ex(E1, E2, NU12, G12, theta) – This function calculates the elastic modulus E_x along the x-direction in the global coordinate system. Its input consists of five arguments representing the four elastic constants E_1, E_2, ν_{12}, G_{12}, and the fiber orientation angle θ.

NUxy(E1, E2, NU12, G12, theta) – This function calculates Poisson's ratio ν_{xy} in the global coordinate system. Its input consists of five arguments representing the four elastic constants E_1, E_2, ν_{12}, G_{12}, and the fiber orientation angle θ.

Ey(E1, E2, NU21, G12, theta) – This function calculates the elastic modulus E_y along the y-direction in the global coordinate system. Its input consists of five arguments representing the four elastic constants E_1, E_2, ν_{21}, G_{12}, and the fiber orientation angle θ.

NUyx(E1, E2, NU21, G12, theta) – This function calculates Poisson's ratio ν_{yx} in the global coordinate system. Its input consists of five arguments representing the four elastic constants E_1, E_2, ν_{21}, G_{12}, and the fiber orientation angle θ.

Gxy(E1, E2, NU12, G12, theta) – This function calculates the shear modulus G_{xy} in the global coordinate system. Its input consists of five arguments representing the four elastic constants E_1, E_2, ν_{12}, G_{12}, and the fiber orientation angle θ.

Etaxyx(Sbar) – This function calculates the coefficient of mutual influence of the second kind $\eta_{xy,x}$. It has one argument – the transformed reduced compliance matrix $[\bar{S}]$.

*Etaxy*y(Sbar) – This function calculates the coefficient of mutual influence of the second kind $\eta_{xy,y}$. It has one argument – the transformed reduced compliance matrix $[\bar{S}]$.

*Etax*xy(Sbar) – This function calculates the coefficient of mutual influence of the first kind $\eta_{x,xy}$. It has one argument – the transformed reduced compliance matrix $[\bar{S}]$.

*Etay*xy(Sbar) – This function calculates the coefficient of mutual influence of the first kind $\eta_{y,xy}$. It has one argument – the transformed reduced compliance matrix $[\bar{S}]$.

The following is a listing of the MATLAB source code for each function:

```
function y = Ex(E1,E2,NU12,G12,theta)
%Ex      This function returns the elastic modulus
%        along the x-direction in the global
%        coordinate system. It has five arguments:
%        E1    - longitudinal elastic modulus
%        E2    - transverse elastic modulus
%        NU12  - Poisson's ratio
%        G12   - shear modulus
%        theta - fiber orientation angle
%        The angle "theta" must be given in degrees.
%        Ex is returned as a scalar
m = cos(theta*pi/180);
n = sin(theta*pi/180);
denom = m^4 + (E1/G12 - 2*NU12)*n*n*m*m + (E1/E2)*n^4;
y = E1/denom;
```

```
function y = NUxy(E1,E2,NU12,G12,theta)
%NUxy   This function returns Poisson’s ratio
%       NUxy in the global
%       coordinate system. It has five arguments:
%       E1    - longitudinal elastic modulus
%       E2    - transverse elastic modulus
%       NU12  - Poisson’s ratio
%       G12   - shear modulus
%       theta - fiber orientation angle
%       The angle "theta" must be given in degrees.
%       NUxy is returned as a scalar
m = cos(theta*pi/180);
n = sin(theta*pi/180);
denom = m^4 + (E1/G12 - 2*NU12)*n*n*m*m + (E1/E2)*n*n;
numer = NU12*(n^4 + m^4) - (1 + E1/E2 - E1/G12)*n*n*m*m;
y = numer/denom;
```

```
function y = Ey(E1,E2,NU21,G12,theta)
%Ey   This function returns the elastic modulus
%     along the y-direction in the global
%     coordinate system. It has five arguments:
%     E1    - longitudinal elastic modulus
%     E2    - transverse elastic modulus
%     NU21  - Poisson’s ratio
%     G12   - shear modulus
%     theta - fiber orientation angle
%     The angle "theta" must be given in degrees.
%     Ey is returned as a scalar
m = cos(theta*pi/180);
n = sin(theta*pi/180);
denom = m^4 + (E2/G12 - 2*NU21)*n*n*m*m + (E2/E1)*n^4;
y = E2/denom;
```

```
function y = NUyx(E1,E2,NU21,G12,theta)
%NUyx   This function returns Poisson’s ratio
%       NUyx in the global
%       coordinate system. It has five arguments:
%       E1    - longitudinal elastic modulus
%       E2    - transverse elastic modulus
%       NU21  - Poisson’s ratio
%       G12   - shear modulus
%       theta - fiber orientation angle
%       The angle "theta" must be given in degrees.
%       NUyx is returned as a scalar
m = cos(theta*pi/180);
n = sin(theta*pi/180);
denom = m^4 + (E2/G12 - 2*NU21)*n*n*m*m + (E2/E1)*n*n;
numer = NU21*(n^4 + m^4) - (1 + E2/E1 - E2/G12)*n*n*m*m;
y = numer/denom;
```

```
function y = Gxy(E1,E2,NU12,G12,theta)
%Gxy This function returns the shear modulus
%    Gxy in the global
%    coordinate system. It has five arguments:
%    E1    - longitudinal elastic modulus
%    E2    - transverse elastic modulus
%    NU12  - Poisson's ratio
%    G12   - shear modulus
%    theta - fiber orientation angle
%    The angle "theta" must be given in degrees.
%    Gxy is returned as a scalar
m = cos(theta*pi/180);
n = sin(theta*pi/180);
denom = n^4 + m^4 + 2*(2*G12*(1 + 2*NU12)/E1 + 2*G12/E2 - 1)
*n*n*m*m;
 y = G12/denom;
```

```
function y = Etaxyx(Sbar)
%Etaxyx   This function returns the coefficient of
%         mutual influence of the second kind
%         ETAxy,x in the global coordinate system.
%         It has one argument - the reduced
%         transformed compliance matrix Sbar.
%         Etaxyx is returned as a scalar
y = Sbar(1,3)/Sbar(1,1);
```

```
function y = Etaxyy(Sbar)
%Etaxyy   This function returns the coefficient of
%         mutual influence of the second kind
%         ETAxy,y in the global coordinate system.
%         It has one argument - the reduced
%         transformed compliance matrix Sbar.
%         Etaxyy is returned as a scalar
y = Sbar(2,3)/Sbar(2,2);
```

```
function y = Etaxxy(Sbar)
%Etaxxy   This function returns the coefficient of
%         mutual influence of the first kind
%         ETAx,xy in the global coordinate system.
%         It has one argument - the reduced
%         transformed compliance matrix Sbar.
%         Etaxxy is returned as a scalar
y = Sbar(1,3)/Sbar(3,3);
```

```
function y = Etayxy(Sbar)
%Etayxy   This function returns the coefficient of
%         mutual influence of the first kind
%         ETAy,xy in the global coordinate system.
%         It has one argument - the reduced
%         transformed compliance matrix Sbar.
%         Etayxy is returned as a scalar
y = Sbar(2,3)/Sbar(3,3);
```

Example 6.1

Derive the expression for E_x given in (6.1).

Solution

From an elementary course on mechanics of materials, we have the following relation (assuming uniaxial tension with $\sigma_x \neq 0$ and all other stresses zeros):

$$\varepsilon_x = \frac{\sigma_x}{E_x} \tag{6.14}$$

However, from (5.10), we also have the following relation:

$$\varepsilon_x = \bar{S}_{11}\sigma_x \tag{6.15}$$

Comparing (6.14) and (6.15), we conclude the following:

$$\frac{1}{E_x} = \bar{S}_{11} \tag{6.16}$$

Substituting for $\bar{S}_{11}$ from (5.16a) and taking the inverse of (6.16), we obtain the desired result as follows:

$$E_x = \frac{1}{\bar{S}_{11}} = \frac{E_1}{m^4 + \left(\frac{E_1}{G_{12}} - 2\nu_{12}\right) n^2 m^2 + \frac{E_1}{E_2} n^4} \tag{6.17}$$

In the above equation, we have substituted for the elements of the reduced compliance matrix with the appropriate elastic constants.

MATLAB Example 6.2

Consider a graphite-reinforced polymer composite lamina with the elastic constants as given in Example 2.2. Use MATLAB to plot the values of the five elastic constants E_x, ν_{xy}, E_y, ν_{yx}, and G_{xy} as a function of the orientation angle θ in the range $-\pi/2 \leq \theta \leq \pi/2$.

Solution

This example is solved using MATLAB. The elastic modulus E_x is calculated at each value of θ between -90° and 90° in increments of 10° using the MATLAB function *Ex*.

```
>> Ex1 = Ex(155.0, 12.10, 0.248, 4.40, -90)

Ex1 =

   12.1000

>> Ex2 = Ex(155.0, 12.10, 0.248, 4.40, -80)

Ex2 =

   11.8632

>> Ex3 = Ex(155.0, 12.10, 0.248, 4.40, -70)

Ex3 =

   11.4059

>> Ex4 = Ex(155.0, 12.10, 0.248, 4.40, -60)

Ex4 =

   11.2480

>> Ex5 = Ex(155.0, 12.10, 0.248, 4.40, -50)

Ex5 =

   11.9204

>> Ex6 = Ex(155.0, 12.10, 0.248, 4.40, -40)

Ex6 =

   14.1524

>> Ex7 = Ex(155.0, 12.10, 0.248, 4.40, -30)

Ex7 =

   19.6820

>> Ex8 = Ex(155.0, 12.10, 0.248, 4.40, -20)
```

```
Ex8 =

   34.1218

>> Ex9 = Ex(155.0, 12.10, 0.248, 4.40, -10)

Ex9 =

   78.7623

>> Ex10 = Ex(155.0, 12.10, 0.248, 4.40, 0)

Ex10 =

   155

>> Ex11 = Ex(155.0, 12.10, 0.248, 4.40, 10)

Ex11 =

   78.7623

>> Ex12 = Ex(155.0, 12.10, 0.248, 4.40, 20)

Ex12 =

   34.1218

>> Ex13 = Ex(155.0, 12.10, 0.248, 4.40, 30)

Ex13 =

   19.6820

>> Ex14 = Ex(155.0, 12.10, 0.248, 4.40, 40)

Ex14 =

   14.1524

>> Ex15 = Ex(155.0, 12.10, 0.248, 4.40, 50)

Ex15 =

   11.9204

>> Ex16 = Ex(155.0, 12.10, 0.248, 4.40, 60)
```

```
Ex16 =

   11.2480

>> Ex17 = Ex(155.0, 12.10, 0.248, 4.40, 70)

Ex17 =

   11.4059

>> Ex18 = Ex(155.0, 12.10, 0.248, 4.40, 80)

Ex18 =

   11.8632

>> Ex19 = Ex(155.0, 12.10, 0.248, 4.40, 90)

Ex19 =

   12.1000
```

The x-axis is now setup for the plots as follows:

```
>> x = [-90 -80 -70 -60 -50 -40 -30 -20 -10 0 10 20 30 40 50 60
        70 80 90]

x =

   -90   -80   -70   -60   -50   -40   -30   -20   -10     0    10   20     30
   40    50    60    70    80    90
```

The values of E_x are now calculated for each value of θ between -90° and 90° in increments of 10°.

```
>> y1 = [Ex1 Ex2 Ex3 Ex4 Ex5 Ex6 Ex7 Ex8 Ex9 Ex10 Ex11 Ex12 Ex13 Ex14
         Ex15 Ex16 Ex17 Ex18 Ex19]

y1 =

   Columns 1 through 14

    12.1000    11.8632    11.4059    11.2480     11.9204    14.1524   19.6820
    34.1218    78.7623   155.0000    78.7623    34.1218    19.6820    14.1524

   Columns 15 through 19

    11.9204    11.2480    11.4059    11.8632    12.1000
```

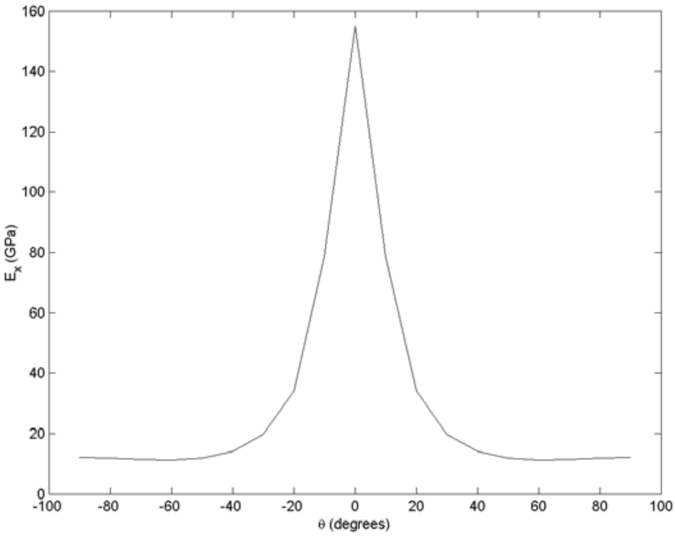

Fig. 6.1. Variation of E_x versus θ for Example 6.2

The plot of the values of E_x versus θ is now generated using the following commands and is shown in Fig. 6.1. Notice that this modulus is an even function of θ. Notice also the rapid variation of the modulus as θ increases or decreases from $0°$.

```
>> plot(x,y1)
>> xlabel('\theta (degrees)');
>> ylabel('E_x (GPa)');
```

Next, Poisson's ratio ν_{xy} is calculated at each value of θ between $-90°$ and $90°$ in increments of $10°$ using the MATLAB function *NUxy*.

```
>> NUxy1 = NUxy(155.0, 12.10, 0.248, 4.40, -90)

NUxy1 =

    0.0194


>> NUxy2 = NUxy(155.0, 12.10, 0.248, 4.40, -80)


NUxy2 =

    0.0640

>> NUxy3 = NUxy(155.0, 12.10, 0.248, 4.40, -70)
```

```
NUxy3 =

    0.1615

>> NUxy4 = NUxy(155.0, 12.10, 0.248, 4.40, -60)

NUxy4 =

    0.2577

>> NUxy5 = NUxy(155.0, 12.10, 0.248, 4.40, -50)

NUxy5 =

    0.3303

>> NUxy6 = NUxy(155.0, 12.10, 0.248, 4.40, -40)

NUxy6 =

    0.3785

>> NUxy7 = NUxy(155.0, 12.10, 0.248, 4.40, -30)

NUxy7 =

    0.4058

>> NUxy8 = NUxy(155.0, 12.10, 0.248, 4.40, -20)

NUxy8 =

    0.4107

>> NUxy9 = NUxy(155.0, 12.10, 0.248, 4.40, -10)

NUxy9 =

    0.3670

>> NUxy10 = NUxy(155.0, 12.10, 0.248, 4.40, 0)

NUxy10 =

    0.2480

>> NUxy11 = NUxy(155.0, 12.10, 0.248, 4.40, 10)
```

```
NUxy11 =

    0.3670

>> NUxy12 = NUxy(155.0, 12.10, 0.248, 4.40, 20)

NUxy12 =

    0.4107

>> NUxy13 = NUxy(155.0, 12.10, 0.248, 4.40, 30)

NUxy13 =

    0.4058

>> NUxy14 = NUxy(155.0, 12.10, 0.248, 4.40, 40)

NUxy14 =

    0.3785

>> NUxy15 = NUxy(155.0, 12.10, 0.248, 4.40, 50)

NUxy15 =

    0.3303

>> NUxy16 = NUxy(155.0, 12.10, 0.248, 4.40, 60)

NUxy16 =

    0.2577

>> NUxy17 = NUxy(155.0, 12.10, 0.248, 4.40, 70)

NUxy17 =

    0.1615

>> NUxy18 = NUxy(155.0, 12.10, 0.248, 4.40, 80)

NUxy18 =

    0.0640

>> NUxy19 = NUxy(155.0, 12.10, 0.248, 4.40, 90)
```

```
NUxy19 =

    0.0194
```

The values of ν_{xy} are now calculated for each value of θ between -90° and 90° in increments of 10°.

```
>> y2 = [NUxy1 NUxy2 NUxy3 NUxy4 NUxy5 NUxy6 NUxy7 NUxy8 NUxy9 NUxy10
        NUxy11 NUxy12 NUxy13 NUxy14 NUxy15 NUxy16 NUxy17 NUxy18 NUxy19]

y2 =

  Columns 1 through 14

    0.0194    0.0640    0.1615    0.2577    0.3303    0.3785
    0.4058    0.4107   0.3670  0.2480  0.3670    0.4107    0.4058    0.3785

  Columns 15 through 19

    0.3303     0.2577     0.1615     0.0640     0.0194
```

The plot of the values of ν_{xy} versus θ is now generated using the following commands and is shown in Fig. 6.2. Notice that this ratio is an even function of θ. Notice also the rapid variation of the ratio as θ increases or decreases from 0°.

```
>> plot(x,y2)
>> xlabel('\theta (degrees)');
>> ylabel('\nu_{xy}');
```

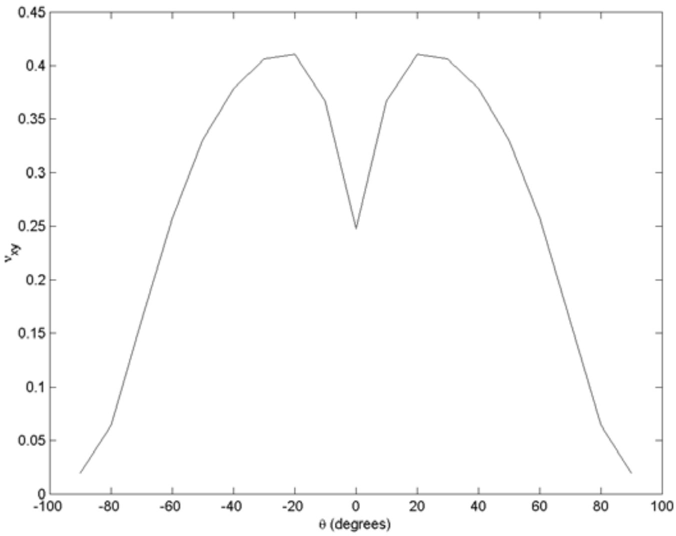

Fig. 6.2. Variation of ν_{xy} versus θ for Example 6.2

Next, the elastic modulus E_y is calculated at each value of θ between -90° and 90° in increments of 10° using the MATLAB function Ey.

```
>> Ey1 = Ey(155.0, 12.10, 0.248, 4.40, -90)

Ey1 =

   155

>> Ey2 = Ey(155.0, 12.10, 0.248, 4.40, -80)

Ey2 =

   86.2721

>> Ey3 = Ey(155.0, 12.10, 0.248, 4.40, -70)

Ey3 =

   39.3653

>> Ey4 = Ey(155.0, 12.10, 0.248, 4.40, -60)

Ey4 =

   22.8718

>> Ey5 = Ey(155.0, 12.10, 0.248, 4.40, -50)

Ey5 =

   16.2611

>> Ey6 = Ey(155.0, 12.10, 0.248, 4.40, -40)

Ey6 =

   13.3820

>> Ey7 = Ey(155.0, 12.10, 0.248, 4.40, -30)

Ey7 =

   12.2222

>> Ey8 = Ey(155.0, 12.10, 0.248, 4.40, -20)
```

```
Ey8 =

   11.9374

>> Ey9 = Ey(155.0, 12.10, 0.248, 4.40, -10)

Ey9 =

   12.0208

>> Ey10 = Ey(155.0, 12.10, 0.248, 4.40, 0)

Ey10 =

   12.1000

>> Ey11 = Ey(155.0, 12.10, 0.248, 4.40, 10)

Ey11 =

   12.0208

>> Ey12 = Ey(155.0, 12.10, 0.248, 4.40, 20)

Ey12 =

   11.9374

>> Ey13 = Ey(155.0, 12.10, 0.248, 4.40, 30)

Ey13 =

   12.2222

>> Ey14 = Ey(155.0, 12.10, 0.248, 4.40, 40)

Ey14 =

   13.3820

>> Ey15 = Ey(155.0, 12.10, 0.248, 4.40, 50)

Ey15 =

   16.2611

>> Ey16 = Ey(155.0, 12.10, 0.248, 4.40, 60)
```

```
Ey16 =

   22.8718

>> Ey17 = Ey(155.0, 12.10, 0.248, 4.40, 70)

Ey17 =

   39.3653

>> Ey18 = Ey(155.0, 12.10, 0.248, 4.40, 80)

Ey18 =

   86.2721

>> Ey19 = Ey(155.0, 12.10, 0.248, 4.40, 90)

Ey19 =

   155
```

The values of E_y are now calculated for each value of θ between -90° and 90° in increments of 10°.

```
>> y3 = [Ey1 Ey2 Ey3 Ey4 Ey5 Ey6 Ey7 Ey8 Ey9 Ey10 Ey11 Ey12 Ey13 Ey14
        Ey15 Ey16 Ey17 Ey18 Ey19]

y3 =

  Columns 1 through 14

   155.0000   86.2721   39.3653   22.8718   16.2611   13.3820 12.2222
   11.9374  12.0208  12.1000  12.0208  11.9374  12.2222  13.3820

  Columns 15 through 19

   16.2611   22.8718   39.3653   86.2721  155.0000
```

The plot of the values of E_y versus θ is now generated using the following commands and is shown in Fig. 6.3. Notice that this modulus is an even function of θ. Notice also the rapid variation of the modulus as θ increases or decreases from 0°.

```
>> plot(x,y3)
>> xlabel('\theta (degrees)');
>> ylabel('E_y (GPa)');
```

Next, Poisson's ratio ν_{yx} is calculated at each value of θ between -90° and 90° in increments of 10° using the MATLAB function *NUyx*

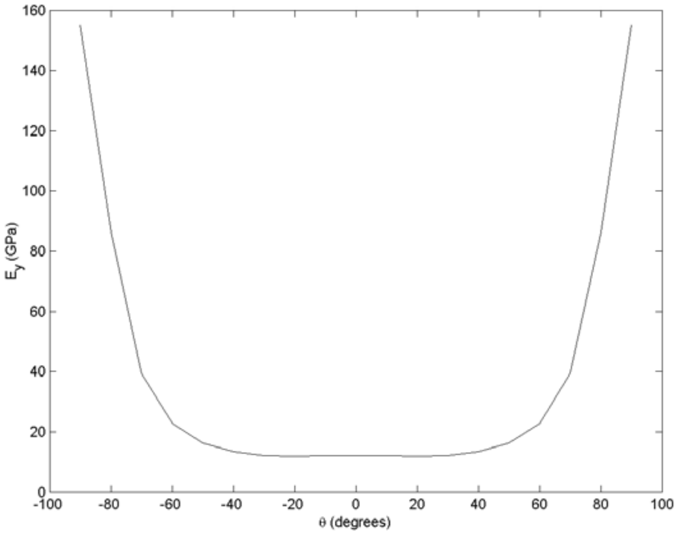

Fig. 6.3. Variation of E_y versus θ for Example 6.2

```
>> NUyx1 = NUyx(155.0, 12.10, 0.248, 4.40, -90)

NUyx1 =

    3.1769

>> NUyx2 = NUyx(155.0, 12.10, 0.248, 4.40, -80)

NUyx2 =

    1.9812

>> NUyx3 = NUyx(155.0, 12.10, 0.248, 4.40, -70)

NUyx3 =

    1.1713

>> NUyx4 = NUyx(155.0, 12.10, 0.248, 4.40, -60)

NUyx4 =

    0.8617

>> NUyx5 = NUyx(155.0, 12.10, 0.248, 4.40, -50)
```

```
NUyx5 =

    0.6987

>> NUyx6 = NUyx(155.0, 12.10, 0.248, 4.40, -40)

NUyx6 =

    0.5775

>> NUyx7 = NUyx(155.0, 12.10, 0.248, 4.40, -30)

NUyx7 =

    0.4663

>> NUyx8 = NUyx(155.0, 12.10, 0.248, 4.40, -20)

NUyx8 =

    0.3616

>> NUyx9 = NUyx(155.0, 12.10, 0.248, 4.40, -10)

NUyx9 =

    0.2799

>> NUyx10 = NUyx(155.0, 12.10, 0.248, 4.40, 0)

NUyx10 =

    0.2480

>> NUyx11 = NUyx(155.0, 12.10, 0.248, 4.40, 10)

NUyx11 =

    0.2799

>> NUyx12 = NUyx(155.0, 12.10, 0.248, 4.40, 20)

NUyx12 =

    0.3616


>> NUyx13 = NUyx(155.0, 12.10, 0.248, 4.40, 30)
```

```
NUyx13 =

    0.4663

>> NUyx14 = NUyx(155.0, 12.10, 0.248, 4.40, 40)

NUyx14 =

    0.5775

>> NUyx15 = NUyx(155.0, 12.10, 0.248, 4.40, 50)

NUyx15 =

    0.6987

>> NUyx16 = NUyx(155.0, 12.10, 0.248, 4.40, 60)

NUyx16 =

    0.8617

>> NUyx17 = NUyx(155.0, 12.10, 0.248, 4.40, 70)

NUyx17 =

    1.1713

>> NUyx18 = NUyx(155.0, 12.10, 0.248, 4.40, 80)

NUyx18 =

    1.9812

>> NUyx19 = NUyx(155.0, 12.10, 0.248, 4.40, 90)

NUyx19 =

    3.1769
```

The values of ν_{yx} are now calculated for each value of θ between -90° and 90° in increments of 10°.

```
>> y4 = [NUyx1 NUyx2 NUyx3 NUyx4 NUyx5 NUyx6 NUyx7 NUyx8 NUyx9 NUyx10
        NUyx11 NUyx12 NUyx13 NUyx14 NUyx15 NUyx16 NUyx17 NUyx18 NUyx19]

y4 =

   Columns 1 through 14
```

```
   3.1769   1.9812   1.1713   0.8617   0.6987   0.5775    0.4663
   0.3616   0.2799   0.2480   0.2799   0.3616   0.4663   0.5775

   Columns 15 through 19

    0.6987    0.8617    1.1713    1.9812    3.1769
```

The plot of the values of ν_{yx} versus θ is now generated using the following commands and is shown in Fig. 6.4. Notice that this ratio is an even function of θ. Notice also the rapid variation of the ratio as θ increases or decreases from $0°$.

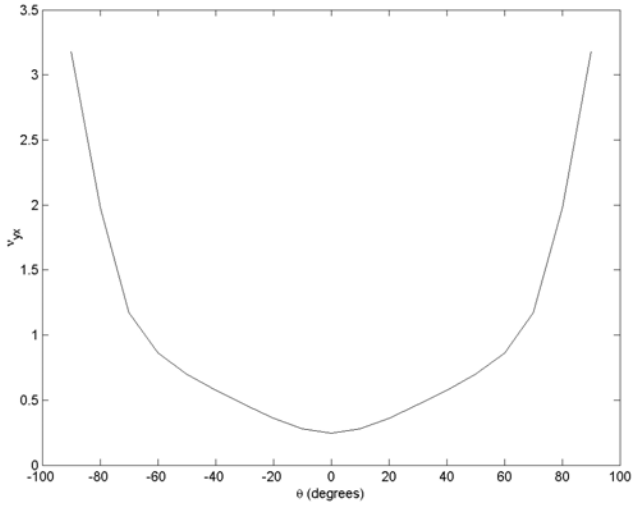

Fig. 6.4. Variation of ν_{yx} versus θ for Example 6.2

```
>> plot(x,y4)
>> xlabel('\theta (degrees)');
>> ylabel('\nu_{yx}');
```

Next, the shear modulus G_{xy} is calculated at each value of θ between $-90°$ and $90°$ in increments of $10°$ using the MATLAB function *Gxy*.

```
>> Gxy1 = Gxy(155.0, 12.10, 0.248, 4.40, -90)

Gxy1 =

    4.4000

>> Gxy2 = Gxy(155.0, 12.10, 0.248, 4.40, -80)

Gxy2 =

    4.7285
```

```
>> Gxy3 = Gxy(155.0, 12.10, 0.248, 4.40, -70)

Gxy3 =

    5.8308

>> Gxy4 = Gxy(155.0, 12.10, 0.248, 4.40, -60)

Gxy4 =

    7.9340

>> Gxy5 = Gxy(155.0, 12.10, 0.248, 4.40, -50)

Gxy5 =

   10.3771

>> Gxy6 = Gxy(155.0, 12.10, 0.248, 4.40, -40)

Gxy6 =

   10.3771

>> Gxy7 = Gxy(155.0, 12.10, 0.248, 4.40, -30)

Gxy7 =

    7.9340

>> Gxy8 = Gxy(155.0, 12.10, 0.248, 4.40, -20)

Gxy8 =

    5.8308

>> Gxy9 = Gxy(155.0, 12.10, 0.248, 4.40, -10)

Gxy9 =

    4.7285

>> Gxy10 = Gxy(155.0, 12.10, 0.248, 4.40, 0)

Gxy10 =

    4.4000
```

```
>> Gxy11 = Gxy(155.0, 12.10, 0.248, 4.40, 10)

Gxy11 =

    4.7285

>> Gxy12 = Gxy(155.0, 12.10, 0.248, 4.40, 20)

Gxy12 =

    5.8308

>> Gxy13 = Gxy(155.0, 12.10, 0.248, 4.40, 30)

Gxy13 =

    7.9340

>> Gxy14 = Gxy(155.0, 12.10, 0.248, 4.40, 40)

Gxy14 =

   10.3771

>> Gxy15 = Gxy(155.0, 12.10, 0.248, 4.40, 50)

Gxy15 =

   10.3771

>> Gxy16 = Gxy(155.0, 12.10, 0.248, 4.40, 60)

Gxy16 =

    7.9340

>> Gxy17 = Gxy(155.0, 12.10, 0.248, 4.40, 70)

Gxy17 =

    5.8308

>> Gxy18 = Gxy(155.0, 12.10, 0.248, 4.40, 80)

Gxy18 =

    4.7285
```

```
>> Gxy19 = Gxy(155.0, 12.10, 0.248, 4.40, 90)

Gxy19 =

    4.4000
```

The values of G_{xy} are now calculated for each value of θ between -90° and 90° in increments of 10°.

```
>> y5 = [Gxy1 Gxy2 Gxy3 Gxy4 Gxy5 Gxy6 Gxy7 Gxy8 Gxy9 Gxy10 Gxy11
        Gxy12  Gxy13 Gxy14 Gxy15 Gxy16 Gxy17 Gxy18 Gxy19]

y5 =

  Columns 1 through 14

   4.4000    4.7285    5.8308    7.9340   10.3771   10.3771    7.9340
   5.8308     4.7285     4.4000     4.7285     5.8308     7.9340   10.3771

  Columns 15 through 19

   10.3771     7.9340     5.8308     4.7285     4.4000
```

The plot of the values of G_{xy} versus θ is now generated using the following commands and is shown in Fig. 6.5. Notice that this modulus is an even function of θ. Notice also the rapid variation of the modulus as θ increases or decreases from 0°.

```
>> plot(x,y5)
>> xlabel('\theta (degrees)');
>> ylabel('G_{xy} (GPa)');
```

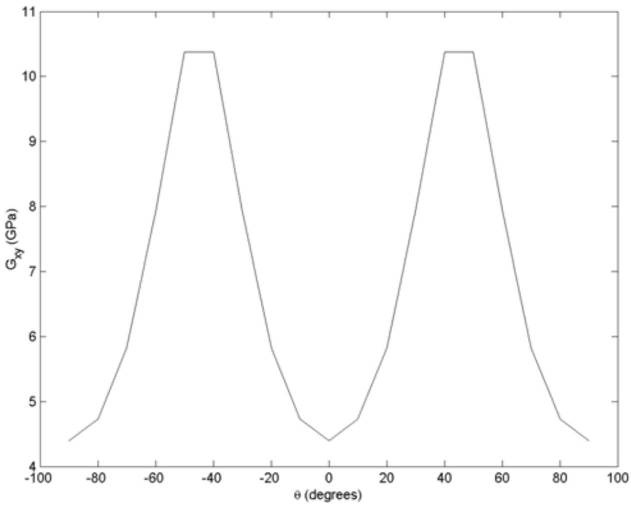

Fig. 6.5. Variation of G_{xy} versus θ for Example 6.2

MATLAB Example 6.3

Consider a graphite-reinforced polymer composite lamina with the elastic constants as given in Example 2.2. Use MATLAB to plot the values of the two coefficients of mutual influence of the second kind $\eta_{xy,x}$ and $\eta_{xy,y}$ as a function of the orientation angle θ in the range $-\pi/2 \leq \theta \leq \pi/2$.

Solution

This example is solved using MATLAB. First, the reduced 3×3 compliance matrix is obtained as follows using the MATLAB function *ReducedCompliance* of Chap. 4.

```
>> S = ReducedCompliance(155.0, 12.10, 0.248, 4.40)

S =

    0.0065   -0.0016         0
   -0.0016    0.0826         0
         0         0    0.2273
```

Next, the transformed reduced compliance matrix $[\bar{S}]$ is calculated at each value of θ between $-90°$ and $90°$ in increments of $10°$ using the MATLAB function *Sbar*.

```
>> S1 = Sbar(S, -90)

S1 =

    0.0826   -0.0016   -0.0000
   -0.0016    0.0065    0.0000
   -0.0000    0.0000    0.2273

>> S2 = Sbar(S, -80)

S2 =

    0.0909   -0.0122   -0.0452
   -0.0122    0.0193    0.0712
   -0.0226    0.0356    0.2061

>> S3 = Sbar(S, -70)

S3 =

    0.1111   -0.0390   -0.0647
   -0.0390    0.0528    0.1137
   -0.0323    0.0568    0.1524

>> S4 = Sbar(S, -60)
```

```
S4 =

    0.1315   -0.0695   -0.0454
   -0.0695    0.0934    0.1114
   -0.0227    0.0557    0.0914

>> S5 = Sbar(S, -50)

S5 =

    0.1390   -0.0894    0.0065
   -0.0894    0.1258    0.0685
    0.0033    0.0342    0.0516

>> S6 = Sbar(S, -40)

S6 =

    0.1258   -0.0894    0.0685
   -0.0894    0.1390    0.0065
    0.0342    0.0033    0.0516

>> S7 = Sbar(S, -30)

S7 =

    0.0934   -0.0695    0.1114
   -0.0695    0.1315   -0.0454
    0.0557   -0.0227    0.0914


>> S8 = Sbar(S, -20)

S8 =

    0.0528   -0.0390    0.1137
   -0.0390    0.1111   -0.0647
    0.0568   -0.0323    0.1524

>> S9 = Sbar(S, -10)

S9 =

    0.0193   -0.0122    0.0712
   -0.0122    0.0909   -0.0452
    0.0356   -0.0226    0.2061

>> S10 = Sbar(S, 0)
```

```
S10 =

    0.0065   -0.0016         0
   -0.0016    0.0826         0
         0         0    0.2273

>> S11 = Sbar(S, 10)

S11 =

    0.0193   -0.0122   -0.0712
   -0.0122    0.0909    0.0452
   -0.0356    0.0226    0.2061

>> S12 = Sbar(S, 20)

S12 =

    0.0528   -0.0390   -0.1137
   -0.0390    0.1111    0.0647
   -0.0568    0.0323    0.1524

>> S13 = Sbar(S, 30)

S13 =

    0.0934   -0.0695   -0.1114
   -0.0695    0.1315    0.0454
   -0.0557    0.0227    0.0914

>> S14 = Sbar(S, 40)

S14 =

    0.1258   -0.0894   -0.0685
   -0.0894    0.1390   -0.0065
   -0.0342   -0.0033    0.0516

>> S15 = Sbar(S, 50)

S15 =

    0.1390   -0.0894   -0.0065
   -0.0894    0.1258   -0.0685
   -0.0033   -0.0342    0.0516

>> S16 = Sbar(S, 60)
```

```
S16 =

    0.1315   -0.0695    0.0454
   -0.0695    0.0934   -0.1114
    0.0227   -0.0557    0.0914

>> S17 = Sbar(S, 70)

S17 =

    0.1111   -0.0390    0.0647
   -0.0390    0.0528   -0.1137
    0.0323   -0.0568    0.1524

>> S18 = Sbar(S, 80)

S18 =

    0.0909   -0.0122    0.0452
   -0.0122    0.0193   -0.0712
    0.0226   -0.0356    0.2061

>> S19 = Sbar(S, 90)

S19 =

    0.0826   -0.0016    0.0000
   -0.0016    0.0065   -0.0000
    0.0000   -0.0000    0.2273
```

The x-axis is now setup for the plots as follows:

```
>> x = [-90  -80  -70  -60  -50  -40  -30  -20  -10  0  10  20  30
       40  50  60  70  80  90]

x =

  -90   -80   -70   -60   -50   -40   -30   -20   -10     0    10
  20    30    40    50    60    70    80    90
```

The values of the coefficient of mutual influence of the second kind $\eta_{xy,x}$ is calculated next for each value of θ in increments of 10° using the MATLAB function *Etaxyx*.

```
>> Etaxyx1 = Etaxyx(S1)

Etaxyx1 =

   -2.1194e-016

>> Etaxyx2 = Etaxyx(S2)
```

```
Etaxyx2 =

   -0.4968

>> Etaxyx3 = Etaxyx(S3)

Etaxyx3 =

   -0.5821

>> Etaxyx4 = Etaxyx(S4)

Etaxyx4 =

   -0.3455

>> Etaxyx5 = Etaxyx(S5)

Etaxyx5 =

    0.0471

>> Etaxyx6 = Etaxyx(S6)

Etaxyx6 =

    0.5446

>> Etaxyx7 = Etaxyx(S7)

Etaxyx7 =

    1.1927

>> Etaxyx8 = Etaxyx(S8)

Etaxyx8 =

    2.1536

>> Etaxyx9 = Etaxyx(S9)

Etaxyx9 =

    3.6831

>> Etaxyx10 = Etaxyx(S10)
```

```
Etaxyx10 =

     0

>> Etaxyx11 = Etaxyx(S11)

Etaxyx11 =

   -3.6831

>> Etaxyx12 = Etaxyx(S12)

Etaxyx12 =

   -2.1536

>> Etaxyx13 = Etaxyx(S13)

Etaxyx13 =

   -1.1927

>> Etaxyx14 = Etaxyx(S14)

Etaxyx14 =

   -0.5446

>> Etaxyx15 = Etaxyx(S15)

Etaxyx15 =

   -0.0471

>> Etaxyx16 = Etaxyx(S16)

Etaxyx16 =

    0.3455

>> Etaxyx17 = Etaxyx(S17)

Etaxyx17 =

    0.5821

>> Etaxyx18 = Etaxyx(S18)
```

```
Etaxyx18 =

    0.4968

>> Etaxyx19 = Etaxyx(S19)

Etaxyx19 =

    2.1194e-016
```

The values of $\eta_{xy,x}$ are now calculated for each value of θ between -90° and 90° in increments of 10°.

```
>> y6 = [Etaxyx1 Etaxyx2 Etaxyx3 Etaxyx4 Etaxyx5 Etaxyx6 Etaxyx7
        Etaxyx8 Etaxyx9 Etaxyx10 Etaxyx11 Etaxyx12 Etaxyx13 Etaxyx14
        Etaxyx15 Etaxyx16 Etaxyx17 Etaxyx18 Etaxyx19]

y6 =

  Columns 1 through 14

    -0.0000   -0.4968   -0.5821   -0.3455    0.0471    0.5446
    1.1927    2.1536    3.6831         0   -3.6831   -2.1536   -1.1927
    -0.5446

  Columns 15 through 19

    -0.0471    0.3455    0.5821    0.4968    0.0000
```

The plot of the values of $\eta_{xy,x}$ versus θ is now generated using the following commands and is shown in Fig. 6.6. Notice that this coefficient is an odd function of θ. Notice also the rapid variation of the coefficient as θ increases or decreases from 0°.

```
>> plot(x,y6)
>> xlabel('\theta (degrees)');
>> ylabel('\eta_{xy,x}');
```

The values of the coefficient of mutual influence of the second kind $\eta_{xy,y}$ is calculated next for each value of θ in increments of 10° using the MATLAB function *Etaxyy*.

```
>> Etaxyy1 = Etaxyy(S1)

Etaxyy1 =

  4.1613e-015

>> Etaxyy2 = Etaxyy(S2)
```

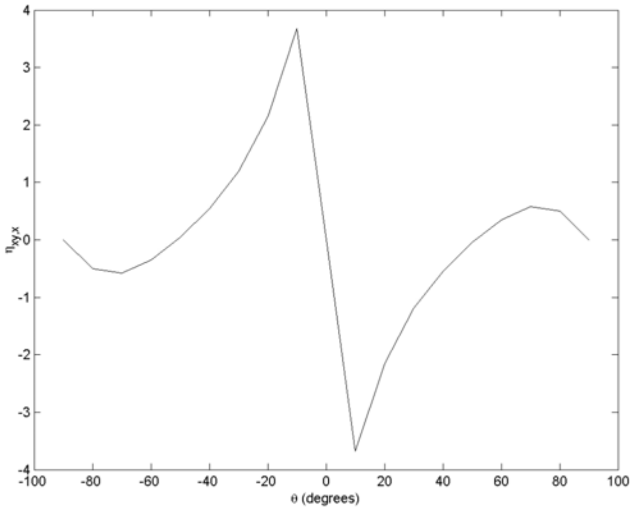

Fig. 6.6. Variation of $\eta_{xy,x}$ versus θ for Example 6.3

```
Etaxyy2 =

    3.6831

>> Etaxyy3 = Etaxyy(S3)

Etaxyy3 =

    2.1536

>> Etaxyy4 = Etaxyy(S4)

Etaxyy4 =

    1.1927

>> Etaxyy5 = Etaxyy(S5)

Etaxyy5 =

    0.5446

>> Etaxyy6 = Etaxyy(S6)

Etaxyy6 =

    0.0471
```

```
>> Etaxyy7 = Etaxyy(S7)

Etaxyy7 =

   -0.3455

>> Etaxyy8 = Etaxyy(S8)

Etaxyy8 =

   -0.5821

>> Etaxyy9 = Etaxyy(S9)

Etaxyy9 =

   -0.4968

>> Etaxyy10 = Etaxyy(S10)

Etaxyy10 =

     0

>> Etaxyy11 = Etaxyy(S11)

Etaxyy11 =

    0.4968

>> Etaxyy12 = Etaxyy(S12)

Etaxyy12 =

    0.5821

>> Etaxyy13 = Etaxyy(S13)

Etaxyy13 =

    0.3455

>> Etaxyy14 = Etaxyy(S14)

Etaxyy14 =

   -0.0471
```

```
>> Etaxyy15 = Etaxyy(S15)

Etaxyy15 =

   -0.5446

>> Etaxyy16 = Etaxyy(S16)

Etaxyy16 =

   -1.1927

>> Etaxyy17 = Etaxyy(S17)

Etaxyy17 =

   -2.1536

>> Etaxyy18 = Etaxyy(S18)

Etaxyy18 =

   -3.6831

>> Etaxyy19 = Etaxyy(S19)

Etaxyy19 =

 -4.1613e-015
```

The values of $\eta_{xy,y}$ are now calculated for each value of θ between $-90°$ and $90°$ in increments of $10°$.

```
>> y7 = [Etaxyy1 Etaxyy2 Etaxyy3 Etaxyy4 Etaxyy5 Etaxyy6 Etaxyy7
        Etaxyy8 Etaxyy9 Etaxyy10 Etaxyy11 Etaxyy12 Etaxyy13 Etaxyy14
        Etaxyy15 Etaxyy16 Etaxyy17 Etaxyy18 Etaxyy19]

y7 =

  Columns 1 through 14

    0.0000    3.6831    2.1536    1.1927    0.5446    0.0471 -0.3455
    -0.5821   -0.4968         0     0.4968    0.5821    0.3455   -0.0471

  Columns 15 through 19

    -0.5446   -1.1927   -2.1536   -3.6831   -0.0000
```

The plot of the values of $\eta_{xy,y}$ versus θ is now generated using the following commands and is shown in Fig. 6.7. Notice that this coefficient is an odd function

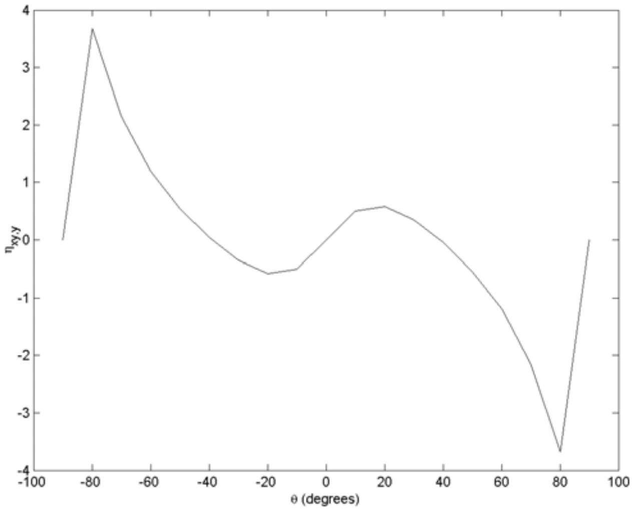

Fig. 6.7. Variation of $\eta_{xy,y}$ versus θ for Example 6.3

of θ. Notice also the rapid variation of the coefficient as θ increases or decreases from 0°.

```
>> plot(x,y7)
>> xlabel('\theta (degrees)');
>> ylabel('\eta_{xy,y}');
```

Problems

Problem 6.1

Derive the expression for ν_{xy} given in (6.2).

Problem 6.2

Derive the expression for E_y given in (6.3).

Problem 6.3

Derive the expression for ν_{yx} given in (6.4).

Problem 6.4

Derive the expression for G_{xy} given in (6.5).

MATLAB Problem 6.5

Consider a glass-reinforced polymer composite lamina with the elastic constants as given in Problem 2.7. Use MATLAB to plot the values of the five elastic constants E_x, ν_{xy}, E_y, ν_{yx}, and G_{xy} as a function of the orientation angle θ in the range $-\pi/2 \leq \theta \leq \pi/2$.

Problem 6.6

Derive the expressions for the coefficients of mutual influence of the second kind $\eta_{xy,x}$ and $\eta_{xy,y}$ given in (6.8) and (6.9).

Problem 6.7

Derive the expressions for the coefficients of mutual influence of the first kind $\eta_{x,xy}$ and $\eta_{y,xy}$ given in (6.12) and (6.13).

MATLAB Problem 6.8

Consider a graphite-reinforced polymer composite lamina with the elastic constants as given in Example 2.2. Use MATLAB to plot the values of the two coefficients of mutual influence of the first kind $\eta_{x,xy}$ and $\eta_{y,xy}$ as a function of the orientation angle θ in the range $-\pi/2 \leq \theta \leq \pi/2$.

MATLAB Problem 6.9

Consider a glass-reinforced polymer composite lamina with the elastic constants as given in Problem 2.7. Use MATLAB to plot the values of the two coefficients of mutual influence of the second kind $\eta_{xy,x}$ and $\eta_{xy,y}$ as a function of the orientation angle θ in the range $-\pi/2 \leq \theta \leq \pi/2$.

MATLAB Problem 6.10

Consider a glass-reinforced polymer composite lamina with the elastic constants as given in Problem 2.7. Use MATLAB to plot the values of the two coefficients of mutual influence of the first kind $\eta_{x,xy}$ and $\eta_{y,xy}$ as a function of the orientation angle θ in the range $-\pi/2 \leq \theta \leq \pi/2$.

7

Laminate Analysis – Part I

7.1 Basic Equations

Fiber-reinforced materials consist usually of multiple layers of material to form a *laminate.* Each layer is thin and may have a different fiber orientation – see Fig. 7.1. Two laminates may have the same number of layers and the same fiber angles but the two laminates may be different because of the arrangement of the layers.

In this chapter, we will evaluate the influence of fiber directions, stacking arrangements and material properties on laminate and structural response. We will study a simplified theory called *classical lamination theory* for this purpose (see [1]).

Figure 7.2 shows a global Cartesian coordinate system and a general laminate consisting of N layers. The laminate thickness is denoted by H and the thickness of an individual layer by h. Not all layers necessarily have the same thickness, so the thickness of the kth layer is denoted by h_k.

The origin of the through-thickness coordinate, designated z, is located at the laminate geometric midplane. The geometric midplane may be within a particular layer or at an interface between layers. We consider the $+z$ axis to be downward and the laminate extends in the z direction from $-H/2$ to $+H/2$. We refer to the layer at the most negative location as layer 1, the next layer in as layer 2, the layer at an arbitrary location as layer k, and the layer at the most positive z position as layer N. The locations of the layer interfaces are denoted by a subscripted z; the

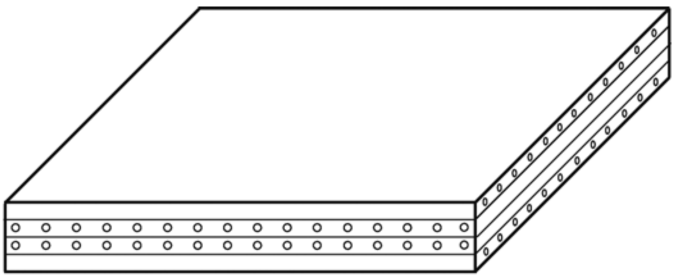

Fig. 7.1. Schematic illustration of a laminate with four layers

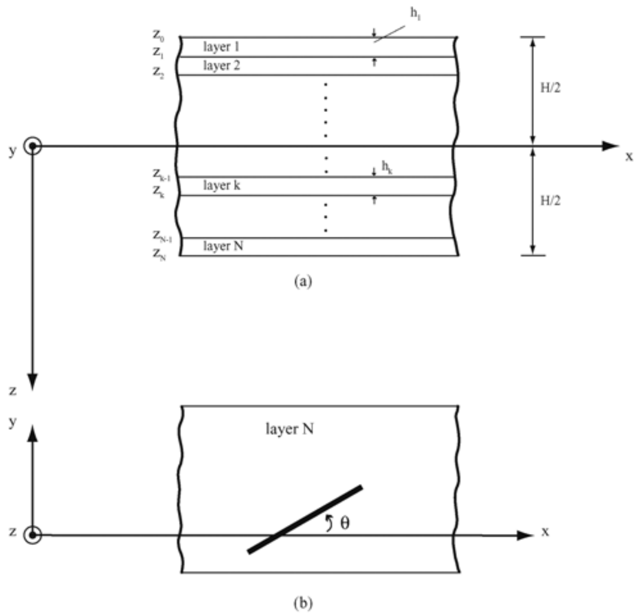

Fig. 7.2. Schematic illustration showing a cross-section and a plan view

first layer is bounded by locations z_0 and z_1, the second layer by z_1 and z_2, the kth layer by z_{k-1} and z_k, and the Nth layer by z_{N-1} and z_N [1].

Let us examine the deformation of an x-z cross-section [1]. Figure 7.3 shows in detail the deformation of a cross-section, and in particular the displacements of point P, a point located at an arbitrary distance z below point P^0, a point on the reference surface, with points P and P^0 being on line AA'. The superscript 0 will be reserved to denote the kinematics of point P^0 on the reference surface. In particular, the horizontal translation of point P^0 in the x direction will be denoted by u^0. The vertical translation will be denoted by w^0. The rotation of the reference surface about the y axis at point P^0 is $\partial w^0/\partial x$. An important part of the Kirchhoff hypothesis is the assumption that line AA' remains perpendicular to the reference surface. Because of this, the rotation of line AA' is the same as the rotation of the reference surface, and thus the rotation of line AA', as viewed in the x-z plane, is $\partial w^0/\partial x$. It is assumed that [1]:

$$\frac{\partial w^0}{\partial x} < 1 \tag{7.1}$$

By less than unity is meant that sines and tangents of angles of rotation are replaced by the rotations themselves, and cosines of the angles of rotation are replace by 1. With this approximation, then, the rotation of point P^0 causes point P to translate horizontally in the minus x direction by an amount equal to:

$$z = \frac{\partial w^0}{\partial x} \tag{7.2}$$

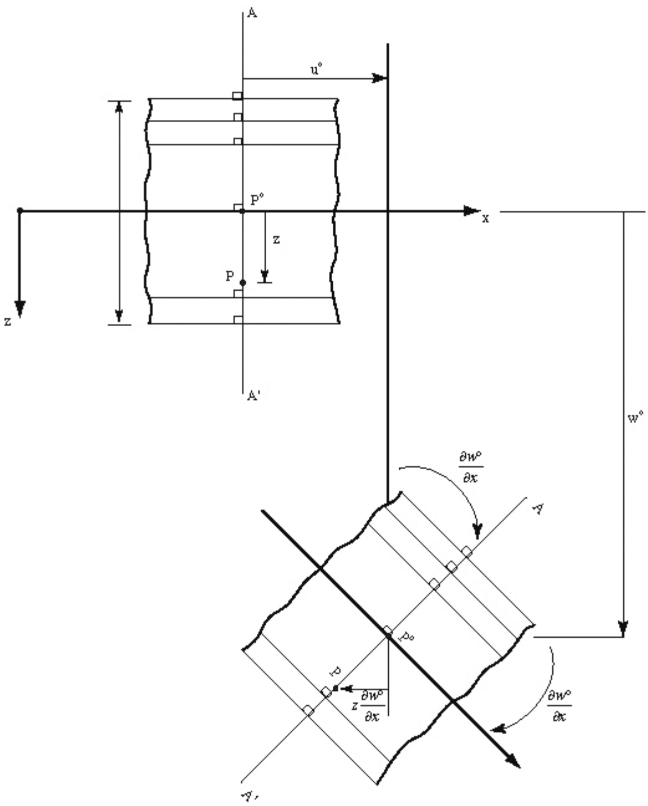

Fig. 7.3. Schematic illustration showing the kinematics of deformation of a laminate

Therefore, the horizontal translation of a point P with coordinates (x, y, z) in the direction of the x-axis is then given by:

$$u(x, y, z) = u^0(x, y) - z\frac{\partial w^0(x, y)}{\partial x} \tag{7.3}$$

Also, the vertical translation of point P in the direction of the z-axis is given by:

$$w(x, y, z) = w^0(x, y) \tag{7.4}$$

The horizontal translation of point P in the direction of the y-axis is similar to that in the direction of the x-axis and is given by:

$$v(x, y, z) = v^0(x, y) - z\frac{\partial w^0(x, y)}{\partial y} \tag{7.5}$$

Therefore, we now have the following relations:

$$u(x, y, z) = u^0(x, y) - z\frac{\partial w^0(x, y)}{\partial x} \tag{7.6a}$$

$$v(x,y,z) = v^0(x,y) - z\frac{\partial w^0(x,y)}{\partial y} \tag{7.6b}$$

$$w(x,y,z) = w^0(x,y) \tag{7.6c}$$

Next, we investigate the strains that result from the displacements according to the Kirchhoff hypothesis. This can be done by using the strain-displacement relations from the theory of elasticity. Using these relations and (7.6a,b,c), we can compute the strains at any point within the laminate, and by using these laminate strains in the stress-strain relations, we can compute the stresses at any point within the laminate.

From the strain-displacement relations and (7.6a), the extensional strain in the x direction, ε_x, is given by:

$$\varepsilon_x(x,y,z) \equiv \frac{\partial u(x,y,z)}{\partial x} = \frac{\partial u^0(x,y)}{\partial x} - z\frac{\partial^2 w^0(x,y)}{\partial x^2} \tag{7.7}$$

Equation (7.7) may be re-written as follows:

$$\varepsilon_x(x,y,z) = \varepsilon_x^0(x,y) + z\kappa_x^0(x,y) \tag{7.8}$$

where the following notation is used:

$$\varepsilon_x^0(x,y) = \frac{\partial u^0(x,y)}{\partial x} \tag{7.9a}$$

$$\kappa_x^0(x,y) = -\frac{\partial^2 w^0(x,y)}{\partial x^2} \tag{7.9b}$$

The quantity ε_x^0 is referred to as the extensional strain of the reference surface in the x direction, and κ_x^0 is referred to as the curvature of the reference surface in the x direction. The other five strain components are given by:

$$\varepsilon_y(x,y,z) \equiv \frac{\partial v(x,y,z)}{\partial y} = \varepsilon_y^0(x,y) + z\kappa_y^0(x,y) \tag{7.10a}$$

$$\varepsilon_z(x,y,z) \equiv \frac{\partial w(x,y,z)}{\partial z} = \frac{\partial w^0(x,y)}{\partial z} = 0 \tag{7.10b}$$

$$\begin{aligned}\gamma_{yz}(x,y,z) &\equiv \frac{\partial w(x,y,z)}{\partial y} + \frac{\partial v(x,y,z)}{\partial z} \\ &= \frac{\partial w^0(x,y)}{\partial y} - \frac{\partial w^0(x,y)}{\partial y} = 0\end{aligned} \tag{7.10c}$$

$$\begin{aligned}\gamma_{xz}(x,y,z) &\equiv \frac{\partial w(x,y,z)}{\partial x} + \frac{\partial u(x,y,z)}{\partial z} \\ &= \frac{\partial w^0(x,y)}{\partial x} - \frac{\partial w^0(x,y)}{\partial x} = 0\end{aligned} \tag{7.10d}$$

$$\gamma_{xy}(x,y,z) \equiv \frac{\partial v(x,y,z)}{\partial x} + \frac{\partial u(x,y,z)}{\partial y} = \gamma_{xy}^0 + z\kappa_{xy}^0 \tag{7.10e}$$

where the following notation is used:

$$\varepsilon_y^0(x,y) = \frac{\partial v^0(x,y)}{\partial y} \tag{7.11a}$$

$$\kappa_y^0(x,y) = -\frac{\partial^2 w^0(x,y)}{\partial y^2} \tag{7.11b}$$

$$\gamma_{xy}^0(x,y) = \frac{\partial v^0(x,y)}{\partial x} + \frac{\partial u^0(x,y)}{\partial y} \tag{7.11c}$$

$$\kappa_{xy}^0(x,y) = -2\frac{\partial^2 w^0(x,y)}{\partial x \partial y} \tag{7.11d}$$

The quantities ε_y^0, κ_y^0, γ_{xy}^0, and κ_{xy}^0 are referred to as the reference surface extensional strain in the y direction, the reference surface curvature in the y direction, the reference surface inplane shear strain, and the reference surface twisting curvature, respectively.

The second important assumption of classical lamination theory is that each point within the volume of a laminate is in a state of plane stress. Therefore, we can compute the stresses if we know the strains and curvatures of the reference surface. Accordingly, using the strains that result from the Kirchhoff hypothesis, (7.8) and (7.10a, e), we find that the stress-strain relations for a laminate become:

$$\left\{ \begin{array}{c} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{array} \right\} = \begin{bmatrix} \bar{Q}_{11} & \bar{Q}_{12} & \bar{Q}_{16} \\ \bar{Q}_{12} & \bar{Q}_{22} & \bar{Q}_{26} \\ \bar{Q}_{16} & \bar{Q}_{26} & \bar{Q}_{66} \end{bmatrix} \left\{ \begin{array}{c} \varepsilon_x^0 + z\kappa_x^0 \\ \varepsilon_y^0 + z\kappa_y^0 \\ \gamma_{xy}^0 + z\kappa_{xy}^0 \end{array} \right\} \tag{7.12}$$

Finally the force and moment resultants in the laminate can be computed using the stresses as follows:

$$N_x = \int_{-H/2}^{H/2} \sigma_x dz \tag{7.13a}$$

$$N_y = \int_{-H/2}^{H/2} \sigma_y dz \tag{7.13b}$$

$$N_{xy} = \int_{-H/2}^{H/2} \tau_{xy} dz \tag{7.13c}$$

$$M_x = \int_{-H/2}^{H/2} \sigma_x z dz \tag{7.13d}$$

$$M_y = \int_{-H/2}^{H/2} \sigma_y z dz \tag{7.13e}$$

$$M_{xy} = \int_{-H/2}^{H/2} \tau_{xy} z dz \tag{7.13f}$$

7.2 MATLAB Functions Used

The only MATLAB function used in this chapter to calculate the strains is:

Strains(eps_xo, eps_yo, gam_xyo, kap_xo, kap_yo, kap_xyo, z) – This function calculates the three strains ε_x, ε_y, and γ_{xy} at any point P on the normal line given the three strains ε_x^0, ε_y^0, γ_{xy}^0 and the three curvatures κ_x^0, κ_y^0, κ_{xy}^0 at point P^0, and the distance z between P and P^0. There are seven input arguments to this function. The function returns the 3×1 strain vector.

The following is a listing of the MATLAB source code for this function:

```
function y = Strains(eps_xo,eps_yo,gam_xyo,kap_xo,kap_yo,kap_xyo,z)
%Strains   This function returns the strain vector at any point P
%          along the normal line at distance z from point Po which
%          lies on the reference surface. There are seven input
%          arguments for this function - namely the three strains
%          and three curvatures at point Po and the distance z.
%          The size of the strain vector is 3 x 1.
epsilonx = eps_xo + z * kap_xo;
epsilony = eps_yo + z * kap_yo;
gammaxy = gam_xyo + z * kap_xyo;
y = [epsilonx ; epsilony ;  gammaxy];
```

MATLAB Example 7.1

Consider a graphite-reinforced polymer composite laminate with the elastic constants as given in Example 2.2. The laminate has total thickness of 0.500 mm and is stacked as a $[0/90]_S$ laminate. The four layers are of equal thickness. It is deformed so that at a point (x, y) on the reference surface, we have the following strains and curvatures:

$$\varepsilon_x^0 = 400 \times 10^{-6}$$
$$\varepsilon_y^0 = \gamma_{xy}^0 = \kappa_x^0 = \kappa_y^0 = \kappa_{xy}^0 = 0$$

Use MATLAB to determine the following:

(a) the three components of strain at the interface locations.
(b) the three components of stress in each layer. Plot the stress distribution along the depth of the laminate for each component.
(c) the force and moment resultants in the laminate.
(d) the three components of strain at the interface locations with respect to the principal material system.
(e) the three components of stress in each layer with respect to the principal material system.

Solution

This example is solved using MATLAB. First the strains are calculated at the five interface locations using the MATLAB function *Strains* as follows:

```
>> epsilon1 = Strains(400e-6,0,0,0,0,0,-0.250e-3)

epsilon1 =

  1.0e-003 *

    0.4000
         0
         0

>> epsilon2 = Strains(400e-6,0,0,0,0,0,-0.125e-3)

epsilon2 =

  1.0e-003 *

    0.4000
         0
         0

>> epsilon3 = Strains(400e-6,0,0,0,0,0,0)

epsilon3 =

  1.0e-003 *

    0.4000
         0
         0

>> epsilon4 = Strains(400e-6,0,0,0,0,0,0.125e-3)

epsilon4 =

  1.0e-003 *

    0.4000
         0
         0

>> epsilon5 = Strains(400e-6,0,0,0,0,0,0.250e-3)

epsilon5 =

  1.0e-003 *

    0.4000
         0
         0
```

Next, the reduced stiffness $[Q\}$ in GPa is calculated for this material using the MATLAB function *ReducedStiffness* as follows:

```
>> Q = ReducedStiffness(155.0, 12.10, 0.248, 4.40)

Q =

  155.7478    3.0153         0
    3.0153   12.1584         0
         0         0    4.4000
```

The transformed reduced stiffnesses $[\bar{Q}]$ in GPa for the four layers are now calculated using the MATLAB function *Qbar* as follows:

```
>> Qbar1 = Qbar(Q,0)

Qbar1 =

  155.7478    3.0153         0
    3.0153   12.1584         0
         0         0    4.4000

>> Qbar2 = Qbar(Q,90)

Qbar2 =

  12.1584    3.0153   -0.0000
   3.0153  155.7478    0.0000
  -0.0000    0.0000    4.4000

>> Qbar3 = Qbar(Q,90)

Qbar3 =

  12.1584    3.0153   -0.0000
   3.0153  155.7478    0.0000
  -0.0000    0.0000    4.4000

>> Qbar4 = Qbar(Q,0)

Qbar4 =

  155.7478    3.0153         0
    3.0153   12.1584         0
         0         0    4.4000
```

Next, the stresses in each layer are calculated in MPa. Note that the stress vector is calculated twice for each layer – once at the top of the layer and once at the bottom of the layer.

```
>> sigma1a = Qbar1*epsilon1*1e3

sigma1a =

   62.2991
    1.2061
         0

>> sigma1b = Qbar1*epsilon2*1e3

sigma1b =

   62.2991
    1.2061
         0

>> sigma2a = Qbar2*epsilon2*1e3

sigma2a =

    4.8634
    1.2061
   -0.0000

>> sigma2b = Qbar2*epsilon3*1e3

sigma2b =

    4.8634
    1.2061
   -0.0000

>> sigma3a = Qbar3*epsilon3*1e3

sigma3a =

    4.8634
    1.2061
   -0.0000

>> sigma3b = Qbar3*epsilon4*1e3

sigma3b =

    4.8634
    1.2061
   -0.0000
```

```
>> sigma4a = Qbar4*epsilon4*1e3

sigma4a =

  62.2991
   1.2061
        0

>> sigma4b = Qbar4*epsilon5*1e3

sigma4b =

  62.2991
   1.2061
        0
```

Next, we setup the y-axis for the three plots:

```
>> y = [0.250 0.125 0.125 0 0 -0.125 -0.125 -0.250]

y =

   0.2500    0.1250    0.1250    0    0    -0.1250    -0.1250
  -0.2500
```

The distribution of the stress σ_x along the depth of the laminate is now plotted as follows (see Fig. 7.4):

```
>> x = [sigma4b(1) sigma4a(1) sigma3b(1) sigma3a(1) sigma2b(1)
       sigma2a(1) sigma1b(1) sigma1a(1)]

x =

  62.2991   62.2991    4.8634    4.8634    4.8634    4.8634   62.2991
  62.2991

>> plot(x,y)
>> xlabel('\sigma_x (MPa)')
>> ylabel('z (mm)')
```

The distribution of the stress σ_y along the depth of the laminate is now plotted as follows (see Fig. 7.5):

```
>> x = [sigma4b(2) sigma4a(2) sigma3b(2) sigma3a(2) sigma2b(2)
       sigma2a(2) sigma1b(2) sigma1a(2)]

x =

   1.2061    1.2061    1.2061    1.2061    1.2061    1.2061    1.2061
   1.2061

>> plot(x,y)
>> ylabel('z (mm)')
>> xlabel('\sigma_y (MPa)')
```

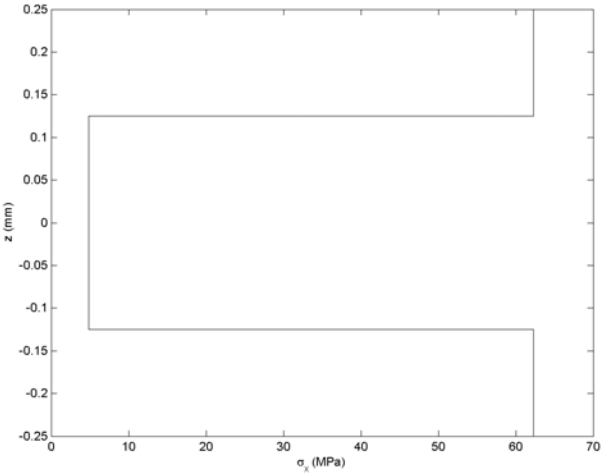

Fig. 7.4. Variation of σ_x versus z for Example 7.1

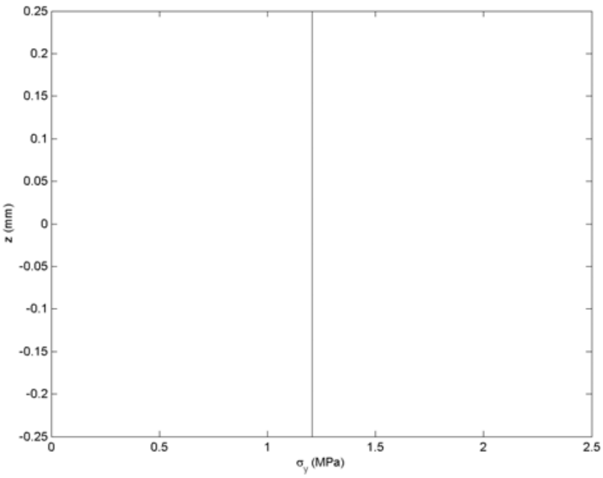

Fig. 7.5. Variation of σ_y versus z for Example 7.1

The distribution of the stress τ_{xy} along the depth of the laminate is now plotted as follows (see Fig. 7.6):

```
>> x = [sigma4b(3) sigma4a(3) sigma3b(3) sigma3a(3) sigma2b(3) sigma2a(3)
sigma1b(3) sigma1a(3)]

x =

  1.0e-015 *

    0    0   -0.1162   -0.1162   -0.1162   -0.1162    0    0
```

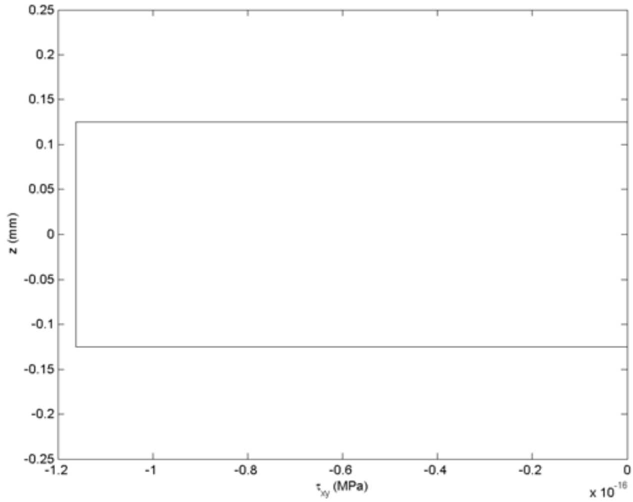

Fig. 7.6. Variation of τ_{xy} versus z for Example 7.1

```
>> plot(x,y)
>> ylabel('z (mm)')
>> xlabel('\tau_{xy} (MPa)')
```

Next, the three force resultants are calculated in MN/m using (7.13a,b,c) as follows:

```
>> Nx = 0.125e-3 * (sigma1a(1) + sigma2a(1) + sigma3a(1) + sigma4a(1))

Nx =

    0.0168

>> Ny = 0.125e-3 * (sigma1a(2) + sigma2a(2) + sigma3a(2) + sigma4a(2))

Ny =

   6.0306e-004

>> Nxy = 0.125e-3 * (sigma1a(3) + sigma2a(3) + sigma3a(3) + sigma4a(3))

Nxy =

   -2.9043e-020
```

Next, the three moment resultants are calculated in MN.m/m using (7.13d,e,f) as follows:

```
>> Mx = sigma1a(1)* ((-0.125e-3)^2 - (-0.250e-3)^2)/2 + sigma2a(1)* (0 -
        (-0.125e-3)^2)/2 + sigma3a(1)* ((0.125e-3)^2 - 0)/2 + sigma4a(1)*
        ((0.250e-3)^2 - (0.125e-3)^2)/2
Mx =

    0

>> My = sigma1a(2)* ((-0.125e-3)^2 - (-0.250e-3)^2)/2 + sigma2a(2)* (0 -
        (-0.125e-3)^2)/2 + sigma3a(2)* ((0.125e-3)^2 - 0)/2 + sigma4a(2)*
        ((0.250e-3)^2 - (0.125e-3)^2)/2

My =

    3.3087e-024

>> Mxy = sigma1a(3)* ((-0.125e-3)^2 - (-0.250e-3)^2)/2 + sigma2a(3)* (0 -
         (-0.125e-3)^2)/2 + sigma3a(3)* ((0.125e-3)^2 - 0)/2 + sigma4a(3)*
         ((0.250e-3)^2 - (0.125e-3)^2)/2

Mxy =

    0
```

Next, the transformation matrix is calculated for each one of the four layers using the MATLAB function T as follows:

```
>> T1 = T(0)

T1 =

     1     0     0
     0     1     0
     0     0     1

>> T2 = T(90)

T2 =

     0.0000    1.0000    0.0000
     1.0000    0.0000   -0.0000
    -0.0000    0.0000   -1.0000

>> T3 = T(90)

T3 =

     0.0000    1.0000    0.0000
     1.0000    0.0000   -0.0000
    -0.0000    0.0000   -1.0000
```

```
>> T4 = T(0)

T4 =

     1     0     0
     0     1     0
     0     0     1
```

The strain vector is now calculated in each layer with respect to the principal material system as follows. Note that the strain vector is calculated twice for each layer – once at the top of the layer and once at the bottom of the layer. Notice also that in this case there is no need to correct the strain vector for the factor of ½ since the shear strain is zero in this example.

```
>> eps1a = T1*epsilon1

eps1a =

    1.0e-003 *

     0.4000
          0
          0

>> eps1b = T1*epsilon2

eps1b =

    1.0e-003 *

     0.4000
          0
          0

>> eps2a = T2*epsilon2

eps2a =

    1.0e-003 *

      0.0000
      0.4000
     -0.0000

>> eps2b = T2*epsilon3
```

```
eps2b =

    1.0e-003 *

       0.0000
       0.4000
      -0.0000

>> eps3a = T3*epsilon3

eps3a =

    1.0e-003 *

       0.0000
       0.4000
      -0.0000

>> eps3b = T3*epsilon4

eps3b =

    1.0e-003 *

       0.0000
       0.4000
      -0.0000

>> eps4a = T4*epsilon4

eps4a =

    1.0e-003 *

      0.4000
           0
           0

>> eps4b = T4*epsilon5

eps4b =

    1.0e-003 *

      0.4000
           0
           0
```

Finally, the stress vector is calculated in MPa for each layer with respect to the principal material systems as follows:

```
>> sig1 = T1*sigma1a

sig1 =

   62.2991
    1.2061
         0

>> sig2 = T2*sigma2a

sig2 =

    1.2061
    4.8634
   -0.0000

>> sig3 = T3*sigma3a

sig3 =

    1.2061
    4.8634
   -0.0000

>> sig4 = T4*sigma4a

sig4 =

   62.2991
    1.2061
         0
```

MATLAB Example 7.2

Consider a graphite-reinforced polymer composite laminate with the elastic constants as given in Example 2.2. The laminate has total thickness of 0.900 mm and is stacked as a $[\pm 30/0]_S$ laminate. The six layers are of equal thickness. It is deformed so that at a point (x, y) on the reference surface, we have the following strains and curvatures:

$$\kappa_x^0 = 2.5\,\mathrm{m}^{-1}$$
$$\varepsilon_x^0 = \varepsilon_y^0 = \gamma_{xy}^0 = \kappa_y^0 = \kappa_{xy}^0 = 0$$

Use MATLAB to determine the following:

(a) the three components of strain at the interface locations.
(b) the three components of stress in each layer. Plot the stress distribution along the depth of the laminate for each component.
(c) the force and moment resultants in the laminate.
(d) the three components of strain at the interface locations with respect to the principal material system.
(e) the three components of stress in each layer with respect to the principal material system.

Solution

This example is solved using MATLAB. First, the strains are calculated at the seven interface locations using the MATLAB function *Strains* as follows:

```
>> epsilon1 = Strains(0,0,0,2.5,0,0,-0.450e-3)

epsilon1 =

   -0.0011
         0
         0

>> epsilon2 = Strains(0,0,0,2.5,0,0,-0.300e-3)

epsilon2 =

   1.0e-003 *

   -0.7500
         0
         0

>> epsilon3 = Strains(0,0,0,2.5,0,0,-0.150e-3)

epsilon3 =

   1.0e-003 *

    -0.3750
          0
          0

>> epsilon4 = Strains(0,0,0,2.5,0,0,0)

epsilon4 =

   0
   0
   0
```

```
>> epsilon5 = Strains(0,0,0,2.5,0,0,0.150e-3)

epsilon5 =

  1.0e-003 *

    0.3750
         0
         0

>> epsilon6 = Strains(0,0,0,2.5,0,0,0.300e-3)

epsilon6 =

  1.0e-003 *

    0.7500
         0
         0

>> epsilon7 = Strains(0,0,0,2.5,0,0,0.450e-3)

epsilon7 =

    0.0011
         0
         0
```

Next, the reduced stiffness $[Q]$ in GPa is calculated for this material using the MATLAB function *ReducedStiffness* as follows:

```
>> Q = ReducedStiffness(155.0, 12.10, 0.248, 4.40)

Q =

   155.7478    3.0153         0
     3.0153   12.1584         0
          0         0    4.4000
```

The transformed reduced stiffnesses $[\bar{Q}]$ in GPa for the six layers are now calculated using the MATLAB function *Qbar* as follows:

```
>> Qbar1 = Qbar(Q,30)

Qbar1 =

   91.1488   31.7170   95.3179
   31.7170   19.3541   29.0342
   47.6589   14.5171   61.8034
```

```
>> Qbar2 = Qbar(Q,-30)

Qbar2 =

   91.1488   31.7170  -95.3179
   31.7170   19.3541  -29.0342
  -47.6589  -14.5171   61.8034

>> Qbar3 = Qbar(Q,0)

Qbar3 =

   155.7478    3.0153         0
     3.0153   12.1584         0
          0         0    4.4000

>> Qbar4 = Qbar(Q,0)

Qbar4 =

   155.7478    3.0153         0
     3.0153   12.1584         0
          0         0    4.4000

>> Qbar5 = Qbar(Q,-30)

Qbar5 =

    91.1488   31.7170  -95.3179
    31.7170   19.3541  -29.0342
   -47.6589  -14.5171   61.8034

>> Qbar6 = Qbar(Q,30)

Qbar6 =
    91.1488   31.7170   95.3179
    31.7170   19.3541   29.0342
    47.6589   14.5171   61.8034
```

Next, the stresses in each layer are calculated in MPa. Note that the stress vector is calculated twice for each layer – once at the top of the layer and once at the bottom of the layer.

```
>> sigma1a = Qbar1*epsilon1*1e3

sigma1a =

   -102.5424
   -35.6816
   -53.6163
```

```
>> sigma1b = Qbar1*epsilon2*1e3

sigma1b =

   -68.3616
   -23.7877
   -35.7442

>> sigma2a = Qbar2*epsilon2*1e3

sigma2a =

   -68.3616
   -23.7877
    35.7442

>> sigma2b = Qbar2*epsilon3*1e3

sigma2b =

   -34.1808
   -11.8939
    17.8721

>> sigma3a = Qbar3*epsilon3*1e3

sigma3a =

   -58.4054
    -1.1307
          0

>> sigma3b = Qbar3*epsilon4*1e3

sigma3b =

     0
     0
     0

>> sigma4a = Qbar4*epsilon4*1e3

sigma4a =

     0
     0
     0
```

```
>> sigma4b = Qbar4*epsilon5*1e3

sigma4b =

    58.4054
     1.1307
          0

>> sigma5a = Qbar5*epsilon5*1e3

sigma5a =

    34.1808
    11.8939
   -17.8721

>> sigma5b = Qbar5*epsilon6*1e3

sigma5b =

    68.3616
    23.7877
   -35.7442

>> sigma6a = Qbar6*epsilon6*1e3

sigma6a =

   68.3616
   23.7877
   35.7442

>> sigma6b = Qbar6*epsilon7*1e3

sigma6b =

   102.5424
    35.6816
    53.6163
```

Next, we setup the y-axis for the three plots:

```
>> y = [0.450 0.300 0.300 0.150 0.150 0 0 -0.150 -0.150 -0.300 -0.300
       -0.450]

y =

    0.4500    0.3000    0.3000    0.1500    0.1500    0    0
    -0.1500   -0.1500   -0.3000   -0.3000   -0.4500
```

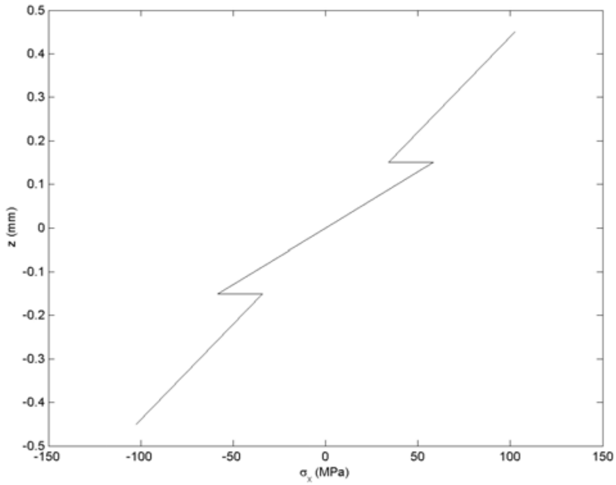

Fig. 7.7. Variation of σ_x versus z for Example 7.2

The distribution of the stress σ_x along the depth of the laminate is now plotted as follows (see Fig. 7.7):

```
>> x = [sigma6b(1) sigma6a(1) sigma5b(1) sigma5a(1) sigma4b(1)
       sigma4a(1) sigma3b(1) sigma3a(1) sigma2b(1) sigma2a(1)
       sigma1b(1) sigma1a(1)]

x =

   102.5424   68.3616   68.3616   34.1808   58.4054         0        0
   -58.4054  -34.1808  -68.3616  -68.3616 -102.5424

>> plot(x,y)
>> xlabel('\sigma_x (MPa)')
>> ylabel('z (mm)')
```

The distribution of the stress σ_y along the depth of the laminate is now plotted as follows (see Fig. 7.8):

```
>> x = [sigma6b(2) sigma6a(2) sigma5b(2) sigma5a(2) sigma4b(2)
       sigma4a(2) sigma3b(2) sigma3a(2) sigma2b(2) sigma2a(2)
       sigma1b(2) sigma1a(2)]

x =

   35.6816   23.7877   23.7877   11.8939   1.1307   0   0
   -1.1307   -11.8939   -23.7877   -23.7877   -35.6816

>> plot(x,y)
>> ylabel('z (mm)')
>> xlabel('\sigma_y (MPa)')
```

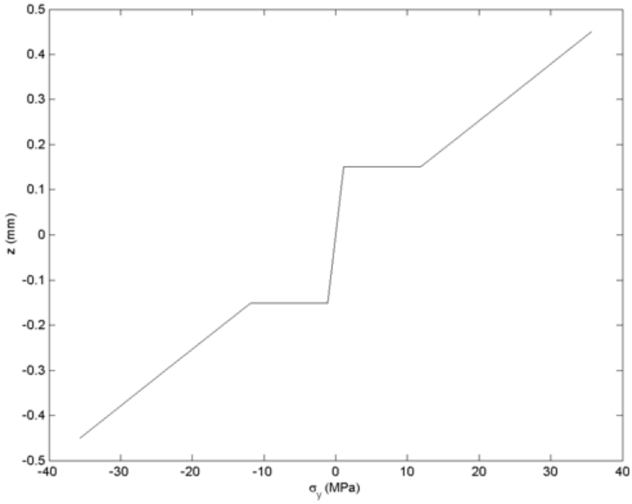

Fig. 7.8. Variation of σ_y versus z for Example 7.2

The distribution of the stress τ_{xy} along the depth of the laminate is now plotted as follows (see Fig. 7.9):

```
>> x = [sigma6b(3) sigma6a(3) sigma5b(3) sigma5a(3) sigma4b(3)
       sigma4a(3) sigma3b(3) sigma3a(3) sigma2b(3) sigma2a(3)
       sigma1b(3) sigma1a(3)]

x =

   53.6163   35.7442   -35.7442   -17.8721   0   0   0
   0   17.8721   35.7442   -35.7442   -53.6163

>> plot(x,y)
>> ylabel('z (mm)')
>> xlabel('\tau_{xy} (MPa)')
```

Next, the three force resultants are calculated in MN/m using (7.13a,b,c) as follows:

```
>> Nx = 0.150 * (sigma1a(1) + sigma2a(1) + sigma3a(1) + sigma4a(1) +
        sigma5a(1) + sigma6a(1))

Nx =

  -19.0150

>> Ny = 0.150 * (sigma1a(2) + sigma2a(2) + sigma3a(2) + sigma4a(2) +
        sigma5a(2) + sigma6a(2))
```

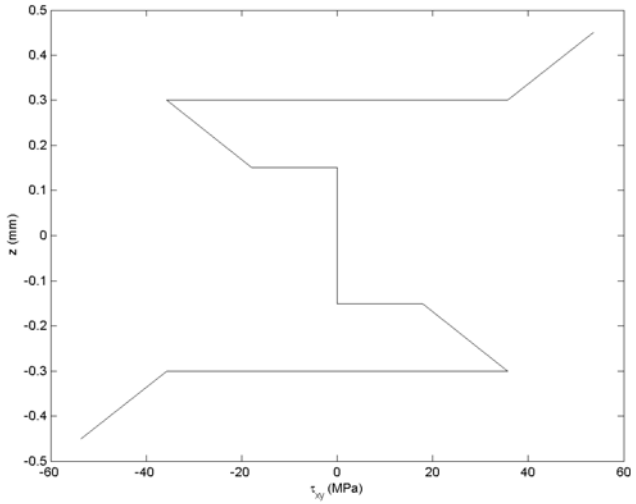

Fig. 7.9. Variation of τ_{xy} versus z for Example 7.2

```
Ny =

   -3.7378

>> Nxy = 0.150 * (sigma1a(3) + sigma2a(3) + sigma3a(3) + sigma4a(3) +
         sigma5a(3) + sigma6a(3))

Nxy =

     0
```

Next, the three moment resultants are calculated in MN.m/m using (7.13d, e, f) as follows:

```
>> Mx = sigma1a(1) * ((-0.300)^2 - (-0.450)^2)/2 + sigma2a(1) *
        (-0.150)^2 - (-0.300)^2)/2 + sigma3a(1) * (0 - (-0.150)^2)/2 +
        sigma4a(1) * ((0.150)^2 - 0)/2 + sigma5a(1) * ((0.300)^2 -
        (0.150)^2)/2 + sigma6a(1) * ((0.450)^2 - (0.300)^2)/2

Mx =

   13.7312

>> My = sigma1a(2) * ((-0.300)^2 - (-0.450)^2)/2 + sigma2a(2) *
        ((-0.150)^2 - (-0.300)^2)/2 + sigma3a(2) * (0 - (-0.150)^2)/2 +
        sigma4a(2) * ((0.150)^2 - 0)/2 + sigma5a(2) * ((0.300)^2 -
        (0.150)^2)/2 + sigma6a(2) * ((0.450)^2 - (0.300)^2)/2
```

```
My =

    4.5621

>> Mxy = sigma1a(3) * ((-0.300)^2 - (-0.450)^2)/2 + sigma2a(3) *
         ((-0.150)^2 - (-0.300)^2)/2 + sigma3a(3) * (0 - (-0.150)^2)/2
         + sigma4a(3) * ((0.150)^2 - 0)/2 + sigma5a(3) * ((0.300)^2 -
         (0.150)^2)/2 + sigma6a(3) * ((0.450)^2 - (0.300)^2)/2

Mxy =

    3.2170
```

Next, the transformation matrix is calculated for each one of the six layers using the MATLAB function T as follows:

```
>> T1 = T(30)

T1 =

    0.7500    0.2500    0.8660
    0.2500    0.7500   -0.8660
   -0.4330    0.4330    0.5000

>> T2 = T(-30)

T2 =

    0.7500    0.2500   -0.8660
    0.2500    0.7500    0.8660
    0.4330   -0.4330    0.5000

>> T3 = T(0)

T3 =

     1     0     0
     0     1     0
     0     0     1

>> T4 = T(0)

T4 =

     1     0     0
     0     1     0
     0     0     1
```

```
>> T5 = T(-30)

T5 =

    0.7500    0.2500   -0.8660
    0.2500    0.7500    0.8660
    0.4330   -0.4330    0.5000

>> T6 = T(30)

T6 =

    0.7500    0.2500    0.8660
    0.2500    0.7500   -0.8660
   -0.4330    0.4330    0.5000
```

The strain vector is now calculated in each layer with respect to the principal material system. Note that the strain vector is calculated twice for each layer – once at the top of the layer and once at the bottom of the layer.

```
>> eps1a = T1*epsilon1

eps1a =

  1.0e-003 *

   -0.8438
   -0.2812
    0.4871

>> eps1b = T1*epsilon2

eps1b =

  1.0e-003 *

   -0.5625
   -0.1875
    0.3248

>> eps2a = T2*epsilon2

eps2a =

  1.0e-003 *

   -0.5625
   -0.1875
   -0.3248
```

```
>> eps2b = T2*epsilon3

eps2b =

  1.0e-003 *

   -0.2813
   -0.0937
   -0.1624

>> eps3a = T3*epsilon3

eps3a =

  1.0e-003 *

   -0.3750
         0
         0

>> eps3b = T3*epsilon4

eps3b =

     0
     0
     0

>> eps4a = T4*epsilon4

eps4a =

     0
     0
     0

>> eps4b = T4*epsilon5

eps4b =

  1.0e-003 *

    0.3750
         0
         0

>> eps5a = T5*epsilon5
```

```
eps5a =

  1.0e-003 *

    0.2813
    0.0937
    0.1624

>> eps5b = T5*epsilon6

eps5b =

  1.0e-003 *

    0.5625
    0.1875
    0.3248

>> eps6a = T6*epsilon6

eps6a =

  1.0e-003 *

    0.5625
    0.1875
   -0.3248

>> eps6b = T6*epsilon7

eps6b =

  1.0e-003 *

    0.8438
    0.2812
   -0.4871
```

Next, we correct the shear strain component for the factor of $1/2$ that appears in the equations.

```
>> eps1a(3) = eps1a(3)*2

eps1a =

  1.0e-003 *

   -0.8438
   -0.2812
    0.9743
```

```
>> eps2a(3) = eps2a(3)*2

eps2a =

  1.0e-003 *

   -0.5625
   -0.1875
   -0.6495

>> eps3a(3) = eps3a(3)*2

eps3a =

  1.0e-003 *

   -0.3750
         0
         0

>> eps4a(3) = eps4a(3)*2

eps4a =

     0
     0
     0

>> eps5a(3) = eps5a(3)*2

eps5a =

  1.0e-003 *

    0.2813
    0.0937
    0.3248

>> eps6a(3) = eps6a(3)*2

eps6a =

  1.0e-003 *

    0.5625
    0.1875
   -0.6495
```

Finally, the stress vector is calculated in MPa for each layer with respect to the principal material system as follows:

```
>> sig1 = T1*sigma1a
sig1 =

 -132.2602
   -5.9637
    2.1434

>> sig2 = T2*sigma2a

sig2 =

  -88.1735
   -3.9758
   -1.4289

>> sig3 = T3*sigma3a

sig3 =

  -58.4054
   -1.1307
         0

>> sig4 = T4*sigma4a

sig4 =

     0
     0
     0

>> sig5 = T5*sigma5a

sig5 =

   44.0867
    1.9879
    0.7145

>> sig6 = T6*sigma6a

sig6 =

   88.1735
    3.9758
   -1.4289
```

Problems

MATLAB Problem 7.1

Consider a glass-reinforced polymer composite laminate with the elastic constants as given in Problem 2.7. The laminate has total thickness of 0.600 mm and is stacked as a $[0/90]_S$ laminate. The four layers are of equal thickness. It is deformed so that at a point (x, y) on the reference surface, we have the following strains and curvatures:

$$\varepsilon_x^0 = 500 \times 10^{-6}$$
$$\varepsilon_y^0 = \gamma_{xy}^0 = \kappa_x^0 = \kappa_y^0 = \kappa_{xy}^0 = 0$$

Use MATLAB to determine the following:

(a) the three components of strain at the interface locations.
(b) the three components of stress in each layer. Plot the stress distribution along the depth of the laminate for each component.
(c) the force and moment resultants in the laminate.
(d) the three components of strain at the interface locations with respect to the principal material system.
(e) the three components of stress in each layer with respect to the principal material system.

MATLAB Problem 7.2

Consider a graphite-reinforced polymer composite laminate with the elastic constants as given in Example 2.2. The laminate has total thickness of 0.600 mm and is stacked as a $[0/90]_S$ laminate. The four layers are of equal thickness. It is deformed so that at a point (x, y) on the reference surface, we have the following strains and curvatures:

$$\kappa_x^0 = 2.5\,\mathrm{m}^{-1}$$
$$\varepsilon_x^0 = \varepsilon_y^0 = \gamma_{xy}^0 = \kappa_y^0 = \kappa_{xy}^0 = 0$$

Use MATLAB to determine the following:

(a) the three components of strain at the interface locations.
(b) the three components of stress in each layer. Plot the stress distribution along the depth of the laminate for each component.
(c) the force and moment resultants in the laminate.
(d) the three components of strain at the interface locations with respect to the principal material system.
(e) the three components of stress in each layer with respect to the principal material system.

MATLAB Problem 7.3

Consider a glass-reinforced polymer composite laminate with the elastic constants as given in Problem 2.7. The laminate has total thickness of 0.600 mm and is stacked as a $[0/90]_S$ laminate. The four layers are of equal thickness. It is deformed so that at a point (x, y) on the reference surface, we have the following strains and curvatures:

$$\kappa_x^0 = 2.5\,\mathrm{m}^{-1}$$
$$\varepsilon_x^0 = \varepsilon_y^0 = \gamma_{xy}^0 = \kappa_y^0 = \kappa_{xy}^0 = 0$$

Use MATLAB to determine the following:

(a) the three components of strain at the interface locations.
(b) the three components of stress in each layer. Plot the stress distribution along the depth of the laminate for each component.
(c) the force and moment resultants in the laminate.
(d) the three components of strain at the interface locations with respect to the principal material system.
(e) the three components of stress in each layer with respect to the principal material system.

MATLAB Problem 7.4

Consider a glass-reinforced polymer composite laminate with the elastic constants as given in Problem 2.7. The laminate has total thickness of 0.900 mm and is stacked as a $[\pm 30/0]_S$ laminate. The six layers are of equal thickness. It is deformed so that at a point (x, y) on the reference surface, we have the following strains and curvatures:

$$\kappa_x^0 = 2.5\,\mathrm{m}^{-1}$$
$$\varepsilon_x^0 = \varepsilon_y^0 = \gamma_{xy}^0 = \kappa_y^0 = \kappa_{xy}^0 = 0$$

Use MATLAB to determine the following:

(a) the three components of strain at the interface locations.
(b) the three components of stress in each layer. Plot the stress distribution along the depth of the laminate for each component.
(c) the force and moment resultants in the laminate.
(d) the three components of strain at the interface locations with respect to the principal material system.
(e) the three components of stress in each layer with respect to the principal material system.

MATLAB Problem 7.5

Consider a graphite-reinforced polymer composite laminate with the elastic constants as given in Example 2.2. The laminate has total thickness of 0.900 mm and is stacked as a $[\pm 30/0]_S$ laminate. The six layers are of equal thickness. It is deformed so that at a point (x, y) on the reference surface, we have the following strains and curvatures:

$$\varepsilon_x^0 = 1000 \times 10^{-6}$$
$$\varepsilon_y^0 = \gamma_{xy}^0 = \kappa_x^0 = \kappa_y^0 = \kappa_{xy}^0 = 0$$

Use MATLAB to determine the following:

(a) the three components of strain at the interface locations.
(b) the three components of stress in each layer. Plot the stress distribution along the depth of the laminate for each component.
(c) the force and moment resultants in the laminate.
(d) the three components of strain at the interface locations with respect to the principal material system.
(e) the three components of stress in each layer with respect to the principal material system.

MATLAB Problem 7.6

Consider a glass-reinforced polymer composite laminate with the elastic constants as given in Problem 2.7. The laminate has total thickness of 0.900 mm and is stacked as a $[\pm 30/0]_S$ laminate. The six layers are of equal thickness. It is deformed so that at a point (x, y) on the reference surface, we have the following strains and curvatures:

$$\varepsilon_x^0 = 1000 \times 10^{-6}$$
$$\varepsilon_y^0 = \gamma_{xy}^0 = \kappa_x^0 = \kappa_y^0 = \kappa_{xy}^0 = 0$$

Use MATLAB to determine the following:

(a) the three components of strain at the interface locations.
(b) the three components of stress in each layer. Plot the stress distribution along the depth of the laminate for each component.
(c) the force and moment resultants in the laminate.
(d) the three components of strain at the interface locations with respect to the principal material system.
(e) the three components of stress in each layer with respect to the principal material system.

8

Laminate Analysis – Part II

8.1 Basic Equations

In Chap. 7, we derived the necessary formulas to calculate the strains and stresses through the thickness and the force and moment resultants given the strains and curvatures at a point (x, y) on the reference surface. In this chapter, we will study the reverse process. Given the force and moment resultants, we want to calculate the stresses and strains through the thickness as well as the strains and curvatures on the reference surface. We also want to do this by computing the laminate stiffness matrix.

Figures 8.1 and 8.2 show the force and moment resultants, respectively. In the two figures, a small element of laminate surrounding a point (x, y) on the geometric midplane is shown [1].

The force resultants N_x, N_y, and N_{xy} can be shown to be related to the strains and curvatures at the reference surface by the following equation:

$$\left\{ \begin{array}{c} N_x \\ N_y \\ N_{xy} \end{array} \right\} = \begin{bmatrix} A_{11} & A_{12} & A_{16} \\ A_{12} & A_{22} & A_{26} \\ A_{16} & A_{26} & A_{66} \end{bmatrix} \left\{ \begin{array}{c} \varepsilon_x^0 \\ \varepsilon_y^0 \\ \gamma_{xy}^0 \end{array} \right\} + \begin{bmatrix} B_{11} & B_{12} & B_{16} \\ B_{12} & B_{22} & B_{26} \\ B_{16} & B_{26} & B_{66} \end{bmatrix} \left\{ \begin{array}{c} \kappa_x^0 \\ \kappa_y^0 \\ \kappa_{XY}^0 \end{array} \right\} \tag{8.1}$$

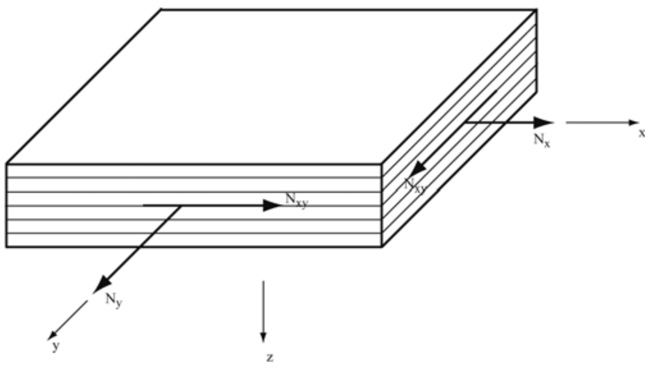

Fig. 8.1. Schematic illustration of the force resultants on a composite laminate

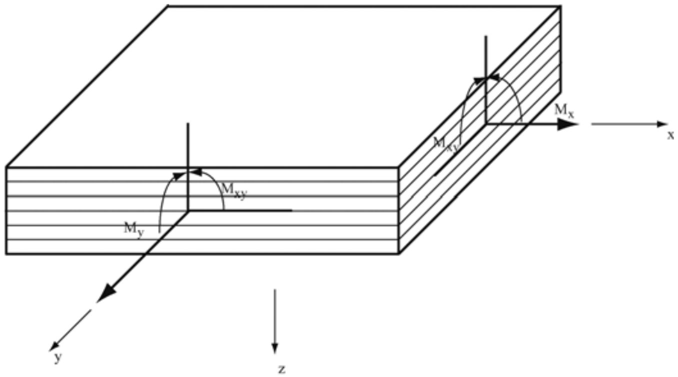

Fig. 8.2. Schematic illustration of the moment resultants on a composite laminate

Similarly, the moment resultants M_x, M_y, and M_{xy} can also be shown to be related to the strains and curvatures at the reference surface by the following equation:

$$\begin{Bmatrix} M_x \\ M_y \\ M_{xy} \end{Bmatrix} = \begin{bmatrix} B_{11} & B_{12} & B_{16} \\ B_{12} & B_{22} & B_{26} \\ B_{16} & B_{26} & B_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_x^0 \\ \varepsilon_y^0 \\ \gamma_{xy}^0 \end{Bmatrix} + \begin{bmatrix} D_{11} & D_{12} & D_{16} \\ D_{12} & D_{22} & D_{26} \\ D_{16} & D_{26} & D_{66} \end{bmatrix} \begin{Bmatrix} \kappa_x^0 \\ \kappa_y^0 \\ \kappa_{XY}^0 \end{Bmatrix} \tag{8.2}$$

where the matrix components A_{ij}, B_{ij}, and D_{ij} are given as follows:

$$A_{ij} = \sum_{k=1}^{N} \bar{Q}_{ij_k} (z_k - z_{k-1}) \tag{8.3}$$

$$B_{ij} = \frac{1}{2} \sum_{k=1}^{N} \bar{Q}_{ij_k} \left(z_k^2 - z_{k-1}^2 \right) \tag{8.4}$$

$$D_{ij} = \frac{1}{3} \sum_{k=1}^{N} \bar{Q}_{ij_k} \left(z_k^3 - z_{k-1}^3 \right) \tag{8.5}$$

Equations (8.1) and (8.2) can be combined into one single equation as follows:

$$\begin{Bmatrix} N_x \\ N_y \\ N_{xy} \\ M_x \\ M_y \\ M_{xy} \end{Bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{16} & B_{11} & B_{12} & B_{16} \\ A_{12} & A_{22} & A_{26} & B_{12} & B_{22} & B_{26} \\ A_{16} & A_{26} & A_{66} & B_{16} & B_{26} & B_{66} \\ B_{11} & B_{12} & B_{16} & D_{11} & D_{12} & D_{16} \\ B_{12} & B_{22} & B_{26} & D_{12} & D_{22} & D_{26} \\ B_{16} & B_{26} & B_{66} & D_{16} & D_{26} & D_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_x^0 \\ \varepsilon_y^0 \\ \gamma_{xy}^0 \\ \kappa_x^0 \\ \kappa_y^0 \\ \kappa_{xy}^0 \end{Bmatrix} \tag{8.6}$$

where the 6×6 matrix consisting of the components A_{ij}, B_{ij}, and D_{ij} $(i,j = 1, 2, 6)$ is called the *laminate stiffness matrix*, sometimes also called the *ABD matrix*. Note that the matrix components A_{ij}, B_{ij}, and D_{ij} represent smeared or integrated properties of the laminate – this is because they are integrals (see [1]).

In order to be able to obtain the strains and curvatures at the reference surface in terms of the force and moment resultants, the inverse of (8.6) is written as follows [1]:

$$\begin{Bmatrix} \varepsilon_x^0 \\ \varepsilon_y^0 \\ \gamma_{xy}^0 \\ \kappa_x^0 \\ \kappa_y^0 \\ \kappa_{xy}^0 \end{Bmatrix} = \begin{Vmatrix} a_{11} & a_{12} & a_{16} & b_{11} & b_{12} & b_{16} \\ a_{12} & a_{22} & a_{26} & b_{12} & b_{22} & b_{26} \\ a_{16} & a_{26} & a_{66} & b_{16} & b_{26} & b_{66} \\ b_{11} & b_{12} & b_{16} & d_{11} & d_{12} & d_{16} \\ b_{12} & b_{22} & b_{26} & d_{12} & d_{22} & d_{26} \\ b_{16} & b_{26} & b_{66} & d_{16} & d_{26} & d_{66} \end{Vmatrix} \begin{Bmatrix} N_x \\ N_y \\ N_{xy} \\ M_x \\ M_y \\ M_{xy} \end{Bmatrix} \tag{8.7}$$

where

$$\begin{bmatrix} a_{11} & a_{12} & a_{16} & b_{11} & b_{12} & b_{16} \\ a_{12} & a_{22} & a_{26} & b_{12} & b_{22} & b_{26} \\ a_{16} & a_{26} & a_{66} & b_{16} & b_{26} & b_{66} \\ b_{11} & b_{12} & b_{16} & d_{11} & d_{12} & d_{16} \\ b_{12} & b_{22} & b_{26} & d_{12} & d_{22} & d_{26} \\ b_{16} & b_{26} & b_{66} & d_{16} & d_{26} & d_{66} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{16} & B_{11} & B_{12} & B_{16} \\ A_{12} & A_{22} & A_{26} & B_{12} & B_{22} & B_{26} \\ A_{16} & A_{26} & A_{66} & B_{16} & B_{26} & B_{66} \\ B_{11} & B_{12} & B_{16} & D_{11} & D_{12} & D_{16} \\ B_{12} & B_{22} & B_{26} & D_{12} & D_{22} & D_{26} \\ B_{16} & B_{26} & B_{66} & D_{16} & D_{26} & D_{66} \end{bmatrix}^{-1} \tag{8.8}$$

Next, we consider the classification of laminates and their effect on the *ABD* matrix. Laminates are usually classified into the following five categories [1]:

1. Symmetric Laminates – A laminate is *symmetric* if for every layer to one side of the laminate reference surface with a specific thickness, specific material properties, and specific fiber orientation, there is another layer the same distance on the opposite side of the reference surface with the same thickness, material properties, and fiber orientation. If the laminate is not symmetric, then it is referred to as an *unsymmetric* laminate.

   For a symmetric laminate, all the components of the B matrix are identically zero. Therefore, we have the following decoupled system of equations:

$$\begin{Bmatrix} N_x \\ N_y \\ N_{xy} \end{Bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{16} \\ A_{12} & A_{22} & A_{26} \\ A_{16} & A_{26} & A_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_x^0 \\ \varepsilon_y^0 \\ \gamma_{xy}^0 \end{Bmatrix} \tag{8.9}$$

$$\begin{Bmatrix} M_x \\ M_y \\ M_{xy} \end{Bmatrix} = \begin{bmatrix} D_{11} & D_{12} & D_{16} \\ D_{12} & D_{22} & D_{26} \\ D_{16} & D_{26} & D_{66} \end{bmatrix} \begin{Bmatrix} \kappa_x^0 \\ \kappa_y^0 \\ \kappa_{XY}^0 \end{Bmatrix} \tag{8.10}$$

2. Balanced Laminates – A laminate is *balanced* if for every layer with a specific thickness, specific material properties, and specific fiber orientation, there is another layer with the same thickness, material properties, but opposite fiber orientation somewhere in the laminate. The other layer can be anywhere within the thickness. For balanced laminates, the stiffness matrix components A_{16} and A_{26} are always zero.
3. Symmetric Balanced Laminates – A laminate is a *symmetric balanced* laminate if it meets both the criterion of being symmetric and the criterion of being balanced. In this case, we have the following decoupled system of equations:

$$\begin{Bmatrix} N_x \\ N_y \end{Bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{12} & A_{22} \end{bmatrix} \begin{Bmatrix} \varepsilon_x^0 \\ \varepsilon_y^0 \end{Bmatrix} \tag{8.11}$$

$$N_{xy} = A_{66}\gamma_{xy}^0 \tag{8.12}$$

$$\begin{Bmatrix} M_x \\ M_y \\ M_{xy} \end{Bmatrix} = \begin{bmatrix} D_{11} & D_{12} & D_{16} \\ D_{12} & D_{22} & D_{26} \\ D_{16} & D_{26} & D_{66} \end{bmatrix} \begin{Bmatrix} \kappa_x^0 \\ \kappa_y^0 \\ \kappa_{XY}^0 \end{Bmatrix} \tag{8.13}$$

4. Cross-Ply Laminates – A laminate is a *cross-ply* laminate if every layer has its fibers oriented at either 0° or 90°. In this case, the components A_{16}, A_{26}, B_{16}, B_{26}, D_{16}, and D_{26} are all zero.

8.2 MATLAB Functions Used

The three MATLAB functions used in this chapter to calculate the $[A]$, $[B]$, and $[D]$ matrices are:

Amatrix(A, Qbar, z1, z2) – This function calculates the $[A]$ matrix for a laminate consisting of N layers where each layer k ($k = 1, 2, 3, \ldots, N$) has a transformed reduced stiffness matrix $[\bar{Q}]^k$. There are four input arguments to this function. This function assembles the desired matrix after each layer's effect is included in a separate call to this function. The parameters $z1$ and $z2$ are z_{k-1} and z_k, respectively, for layer k. The function returns the 3×3 matrix $[A]$.

Bmatrix(B, Qbar, z1, z2) – This function calculates the $[B]$ matrix for a laminate consisting of N layers where each layer k ($k = 1, 2, 3, \ldots, N$) has a transformed reduced stiffness matrix $[\bar{Q}]^k$. There are four input arguments to this function. This function assembles the desired matrix after each layer's effect is included in a separate call to this function. The parameters $z1$ and $z2$ are z_{k-1} and z_k, respectively, for layer k. The function returns the 3×3 matrix $[B]$.

Dmatrix(D, Qbar, z1, z2) – This function calculates the $[D]$ matrix for a laminate consisting of N layers where each layer k ($k = 1, 2, 3, \ldots, N$) has a transformed reduced stiffness matrix $[\bar{Q}]^k$. There are four input arguments to this function. This function assembles the desired matrix after each layer's effect is included in a separate call to this function. The parameters $z1$ and $z2$ are z_{k-1} and z_k, respectively, for layer k. The function returns the 3×3 matrix $[D]$.

The following is a listing of the MATLAB source code for these functions:

```
function y = Amatrix(A,Qbar,z1,z2)
%Amatrix   This function returns the [A] matrix
%          after the layer k with stiffness [Qbar]
%          is assembled.
%          A     - [A] matrix after layer k
%                  is assembled.
%          Qbar  - [Qbar] matrix for layer k
%          z1    - z(k-1) for layer k
%          z2    - z(k) for layer k
```

```
for i = 1 : 3
    for j = 1 : 3
        A(i,j) = A(i,j) + Qbar(i,j)*(z2-z1);
    end
end
y = A;
```

```
function y = Bmatrix(B,Qbar,z1,z2)
%Bmatrix   This function returns the [B] matrix
%          after the layer k with stiffness [Qbar]
%          is assembled.
%          B     - [B] matrix after layer k
%                  is assembled.
%          Qbar  - [Qbar] matrix for layer k
%          z1    - z(k-1) for layer k
%          z2    - z(k) for layer k
for i = 1 : 3
    for j = 1 : 3
        B(i,j) = B(i,j) + Qbar(i,j)*(z2^2 -z1^2);
    end
end
y = B/2;
```

```
function y = Dmatrix(D,Qbar,z1,z2)
%Dmatrix   This function returns the [D] matrix
%          after the layer k with stiffness [Qbar]
%          is assembled.
%          D     - [D] matrix after layer k
%                  is assembled.
%          Qbar  - [Qbar] matrix for layer k
%          z1    - z(k-1) for layer k
%          z2    - z(k) for layer k
for i = 1 : 3
    for j = 1 : 3
        D(i,j) = D(i,j) + Qbar(i,j)*(z2^3 -z1^3);
    end
end
y = D/3;
```

Example 8.1

Derive (8.3) and (8.4) in detail.

Solution

The derivation of (8.3) and (8.4) involves using (7.13a), (7.13b), and (7.13c) along with (7.12). Substitute the expression of σ_x obtained from (7.12) into (7.13a) to obtain:

$$N_x = \int\limits_{-H/2}^{H/2} \left[\bar{Q}_{11}\left(\varepsilon_x^0 + z\kappa_x^0\right) + \bar{Q}_{12}\left(\varepsilon_y^0 + z\kappa_y^0\right) + \bar{Q}_{16}\left(\gamma_{xy}^0 + z\kappa_{xy}^0\right)\right] dz \tag{8.14}$$

Expanding (8.14), we obtain:

$$\begin{aligned} N_x = {} & \varepsilon_x^0 \int\limits_{-H/2}^{H/2} \bar{Q}_{11}\, dz + \kappa_x^0 \int\limits_{-H/2}^{H/2} \bar{Q}_{11}\, zdz + \varepsilon_y^0 \int\limits_{-H/2}^{H/2} \bar{Q}_{12}\, dz + \kappa_y^0 \int\limits_{-H/2}^{H/2} \bar{Q}_{12}\, zdz \\ & + \gamma_{xy}^0 \int\limits_{-H/2}^{H/2} \bar{Q}_{16}\, dz + \kappa_{xy}^0 \int\limits_{-H/2}^{H/2} \bar{Q}_{16}\, zdz \end{aligned} \tag{8.15}$$

Next, we expand the first term of (8.15) as follows:

$$\int\limits_{-H/2}^{H/2} \bar{Q}_{11}\, dz = \int\limits_{z_0}^{z_1} \bar{Q}_{11} dz + \int\limits_{z_1}^{z_2} \bar{Q}_{11} dz + \cdots + \int\limits_{z_{k-1}}^{z_k} \bar{Q}_{11} dz + \cdots + \int\limits_{z_{N-1}}^{z_N} \bar{Q}_{11} dz \tag{8.16}$$

Recognizing that $\bar{Q}_{11}$ is constant within each layer, it can be taken outside the integrals above leading to the following expression:

$$\begin{aligned} \int\limits_{-H/2}^{H/2} \bar{Q}_{11}\, dz = {} & \bar{Q}_{11}\left(z_1 - z_0\right) + \bar{Q}_{11}\left(z_2 - z_1\right) + \cdots + \bar{Q}_{11}\left(z_k - z_{k-1}\right) \\ & + \cdots + \bar{Q}_{11}\left(z_N - z_{N-1}\right) \end{aligned} \tag{8.17}$$

The above equation can be re-written as follows:

$$\int\limits_{-H/2}^{H/2} \bar{Q}_{11}\, dz = \sum_{k=1}^{N} \bar{Q}_{11}\left(z_k - z_{k-1}\right) = A_{11} \tag{8.18}$$

Similarly, we can show that the other five integrals of (8.15) can be written as follows:

$$\int\limits_{-H/2}^{H/2} \bar{Q}_{12}\, dz = \sum_{k=1}^{N} \bar{Q}_{12}\left(z_k - z_{k-1}\right) = A_{12} \tag{8.19a}$$

$$\int\limits_{-H/2}^{H/2} \bar{Q}_{16}\, dz = \sum_{k=1}^{N} \bar{Q}_{16}\left(z_k - z_{k-1}\right) = A_{16} \tag{8.19b}$$

$$\int\limits_{-H/2}^{H/2} \bar{Q}_{11}\, zdz = \frac{1}{2} \sum_{k=1}^{N} \bar{Q}_{11}\left(z_k^2 - z_{k-1}^2\right) = B_{11} \tag{8.19c}$$

$$\int_{-H/2}^{H/2} \bar{Q}_{12}\, z dz = \frac{1}{2} \sum_{k=1}^{N} \bar{Q}_{12} \left(z_k^2 - z_{k-1}^2 \right) = B_{12} \tag{8.19d}$$

$$\int_{-H/2}^{H/2} \bar{Q}_{16}\, z dz = \frac{1}{2} \sum_{k=1}^{N} \bar{Q}_{16} \left(z_k^2 - z_{k-1}^2 \right) = B_{16} \tag{8.19e}$$

Using the remaining two equations of the matrix (7.12), we obtain the general desired expressions as follows:

$$A_{ij} = \sum_{k=1}^{N} \bar{Q}_{ij} \left(z_k - z_{k-1} \right) \tag{8.20}$$

$$B_{ij} = \frac{1}{2} \sum_{k=1}^{N} \bar{Q}_{ij} \left(z_k^2 - z_{k-1}^2 \right) \tag{8.21}$$

MATLAB Example 8.2

Consider a graphite-reinforced polymer composite laminate with the elastic constants as given in Example 2.2. The laminate has total thickness of 0.500 mm and is stacked as a $[0/90]_S$ laminate. The four layers are of equal thickness. Calculate the $[A]$, $[B]$, and $[D]$ matrices for this laminate.

Solution

This example is solved using MATLAB. First, the reduced stiffness matrix $[Q]$ for a typical layer using the MATLAB function *ReducedStiffness* as follows:

```
>> Q = ReducedStiffness(155.0, 12.10, 0.248, 4.40)

Q =

  155.7478    3.0153         0
    3.0153   12.1584         0
         0         0    4.4000
```

Next, the transformed reduced stiffness matrix $[\bar{Q}]$ is calculated for each layer using the MATLAB function *Qbar* as follows:

```
>> Qbar1 = Qbar(Q,0)

Qbar1 =

  155.7478    3.0153         0
    3.0153   12.1584         0
         0         0    4.4000
```

```
>> Qbar2 = Qbar(Q,90)

Qbar2 =

   12.1584    3.0153   -0.0000
    3.0153  155.7478    0.0000
   -0.0000    0.0000    4.4000

>> Qbar3 = Qbar(Q,90)

Qbar3 =

   12.1584    3.0153   -0.0000
    3.0153  155.7478    0.0000
   -0.0000    0.0000    4.4000

>> Qbar4 = Qbar(Q,0)

Qbar4 =

  155.7478    3.0153         0
    3.0153   12.1584         0
         0         0    4.4000
```

Next, the distances $z_k (k = 1, 2, 3, 4, 5)$ are calculated as follows:

```
>> z1 = -0.250

z1 =

   -0.2500

>> z2 = -0.125

z2 =

   -0.1250

>> z3 = 0

z3 =

     0

>> z4 = 0.125

z4 =

    0.1250
```

```
>> z5 = 0.250

z5 =

   0.2500
```

Next, the $[A]$ matrix is calculated using four calls to the MATLAB function *Amatrix* as follows:

```
>> A = zeros(3,3)

A =
     0     0     0
     0     0     0
     0     0     0

>> A = Amatrix(A,Qbar1,z1,z2)

A =

    19.4685    0.3769         0
     0.3769    1.5198         0
          0         0    0.5500

>> A = Amatrix(A,Qbar2,z2,z3)

A =
    20.9883    0.7538   -0.0000
     0.7538   20.9883    0.0000
    -0.0000    0.0000    1.1000

>> A = Amatrix(A,Qbar3,z3,z4)

A =

   22.5081    1.1307   -0.0000
    1.1307   40.4567    0.0000
   -0.0000    0.0000    1.6500

>> A = Amatrix(A,Qbar4,z4,z5)

A =

   41.9765    1.5076   -0.0000
    1.5076   41.9765    0.0000
   -0.0000    0.0000    2.2000
```

Next, the $[B]$ matrix is calculated using four calls to the MATLAB function *Bmatrix* as follows (make sure to divide the final result by 2 since this step is not performed by the *Bmatrix* function):

```
>> B = zeros(3,3)

B =

   0    0    0
   0    0    0
   0    0    0

>> B = Bmatrix(B,Qbar1,z1, z2)

B =

  -7.3007   -0.1413         0
  -0.1413   -0.5699         0
   0              0    -0.2063

>> B = Bmatrix(B,Qbar2,z2, z3)

B =

   -7.4907    -0.1885    0.0000
   -0.1885    -3.0035   -0.0000
    0.0000    -0.0000   -0.2750

>> B = Bmatrix(B,Qbar3,z3, z4)

B =

   -7.3007   -0.1413         0
   -0.1413   -0.5699         0
         0         0    -0.2063

>> B = Bmatrix(B,Qbar4,z4, z5)

B =

  1.0e-015 *

         0          0       0
         0    -0.1110       0
         0          0       0

>> B = B/2

B =

  1.0e-016 *
```

```
         0         0         0
         0   -0.5551         0
         0         0         0
```

Next, the $[D]$ matrix is calculated using four calls to the MATLAB function *Dmatrix* as follows (make sure to divide the final result by 3 since this step is not performed by the *Dmatrix* function):

```
>> D = zeros(3,3)

D =

   0       0       0
   0       0       0
   0       0       0

>> D = Dmatrix(D,Qbar1, z1, z2)

D =

   2.1294      0.0412           0
   0.0412      0.1662           0
        0           0      0.0602

>> D = Dmatrix(D,Qbar2, z2, z3)

D =
   2.1531     0.0471     -0.0000
   0.0471     0.4704      0.0000
  -0.0000     0.0000      0.0688

>> D = Dmatrix(D,Qbar3, z3, z4)

D =

   2.1769     0.0530      -0.0000
   0.0530     0.7746       0.0000
  -0.0000     0.0000       0.0773

>> D = Dmatrix(D,Qbar4, z4, z5)

D =
   4.3062     0.0942     -0.0000
   0.0942     0.9408      0.0000
  -0.0000     0.0000      0.1375
```

```
>> D = D/3

D =
   1.4354    0.0314     -0.0000
   0.0314    0.3136      0.0000
  -0.0000    0.0000      0.0458
```

MATLAB Example 8.3

Consider a graphite-reinforced polymer composite laminate with the elastic constants as given in Example 2.2. The laminate has total thickness of 0.900 mm and is stacked as a $[\pm 30/0]_S$ laminate. The six layers are of equal thickness. Calculate the $[A]$, $[B]$, and $[D]$ matrices for this laminate.

Solution

This example is solved using MATLAB. First, the reduced stiffness matrix $[Q]$ for a typical layer using the MATLAB function *ReducedStiffness* as follows:

```
>> Q = ReducedStiffness(155.0, 12.10, 0.248, 4.40)

Q =

  155.7478    3.0153         0
    3.0153   12.1584         0
         0         0    4.4000
```

Next, the transformed reduced stiffness matrix $[\bar{Q}]$ is calculated for each layer using the MATLAB function *Qbar* as follows:

```
>> Qbar1 = Qbar(Q, 30)

Qbar1 =

   91.1488   31.7170   95.3179
   31.7170   19.3541   29.0342
   47.6589   14.5171   61.8034

>> Qbar2 = Qbar(Q, -30)

Qbar2 =

   91.1488   31.7170  -95.3179
   31.7170   19.3541  -29.0342
  -47.6589  -14.5171   61.8034

>> Qbar3 = Qbar(Q, 0)
```

```
Qbar3 =

  155.7478    3.0153         0
    3.0153   12.1584         0
         0         0    4.4000

>> Qbar4 = Qbar(Q, 0)

Qbar4 =

  155.7478    3.0153         0
    3.0153   12.1584         0
         0         0    4.4000

>> Qbar5 = Qbar(Q, -30)

Qbar5 =

   91.1488   31.7170  -95.3179
   31.7170   19.3541  -29.0342
  -47.6589  -14.5171   61.8034

>> Qbar6 = Qbar(Q, 30)

Qbar6 =

   91.1488   31.7170   95.3179
   31.7170   19.3541   29.0342
   47.6589   14.5171   61.8034
```

Next, the distances z_k $(k = 1, 2, 3, 4, 5, 6, 7)$ are calculated as follows:

```
>> z1 = -0.450

z1 =

   -0.4500

>> z2 = -0.300

z2 =

   -0.3000

>> z3 = -0.150

z3 =

   -0.1500
```

```
>> z4 = 0

z4 =

     0

>> z5 = 0.150

z5 =

    0.1500

>> z6 = 0.300

z6 =

    0.3000

>> z7 = 0.450

z7 =

    0.4500
```

Next, the $[A]$ matrix is calculated using six calls to the MATLAB function *Amatrix* as follows:

```
 >> A = zeros(3,3)

A =

     0     0     0
     0     0     0
     0     0     0

>> A = Amatrix(A,Qbar1,z1,z2)

A =

   13.6723    4.7575   14.2977
    4.7575    2.9031    4.3551
    7.1488    2.1776    9.2705

>> A = Amatrix(A,Qbar2,z2,z3)

A =

   27.3446    9.5151    0.0000
    9.5151    5.8062    0.0000
    0.0000    0.0000   18.5410
```

```
>> A = Amatrix(A,Qbar3,z3,z4)

A =

   50.7068    9.9674    0.0000
    9.9674    7.6300    0.0000
    0.0000    0.0000   19.2010

>> A = Amatrix(A,Qbar4,z4,z5)

A =

   74.0690   10.4197    0.0000
   10.4197    9.4537    0.0000
    0.0000    0.0000   19.8610

>> A = Amatrix(A,Qbar5,z5,z6)

A =

   87.7413   15.1772  -14.2977
   15.1772   12.3568   -4.3551
   -7.1488   -2.1776   29.1315

>> A = Amatrix(A,Qbar6,z6,z7)

A =

  101.4136   19.9348    0.0000
   19.9348   15.2599    0.0000
    0.0000    0.0000   38.4020
```

Next, the $[B]$ matrix is calculated using six calls to the MATLAB function *Bmatrix* as follows (make sure to divide the final result by 2 since this step is not performed by the *Bmatrix* function):

```
>> B = zeros(3,3)
B =

     0     0     0
     0     0     0
     0     0     0

>> B = Bmatrix(B,Qbar1,z1,z2)

B =

  -10.2542   -3.5682  -10.7233
   -3.5682   -2.1773   -3.2663
   -5.3616   -1.6332   -6.9529
```

```
>> B = Bmatrix(B,Qbar2,z2,z3)

B =

  -16.4068   -5.7091   -4.2893
   -5.7091   -3.4837   -1.3065
   -2.1447   -0.6533  -11.1246

>> B = Bmatrix(B,Qbar3,z3,z4)

B =

  -19.9111   -5.7769   -4.2893
   -5.7769   -3.7573   -1.3065
   -2.1447   -0.6533  -11.2236

>> B = Bmatrix(B,Qbar4,z4,z5)

B =

  -16.4068   -5.7091   -4.2893
   -5.7091   -3.4837   -1.3065
   -2.1447   -0.6533  -11.1246

>> B = Bmatrix(B,Qbar5,z5,z6)

B =

  -10.2542   -3.5682  -10.7233
   -3.5682   -2.1773   -3.2663
   -5.3616   -1.6332   -6.9529

>> B = Bmatrix(B,Qbar6,z6,z7)

B =

  1.0e-015 *

         0   -0.4441         0
         0         0         0
         0         0   -0.8882

>> B = B/2

B =

  1.0e-015 *
```

```
         0   -0.2220         0
         0         0         0
         0         0   -0.4441
```

Next, the $[D]$ matrix is calculated using six calls to the MATLAB function *Dmatrix* as follows (make sure to divide the final result by 3 since this step is not performed by the *Dmatrix* function):

```
>> D = zeros(3,3)

D =

     0     0     0
     0     0     0
     0     0     0

>> D = Dmatrix(D,Qbar1,z1,z2)

D =

    5.8449    2.0338    6.1123
    2.0338    1.2411    1.8618
    3.0561    0.9309    3.9631

>> D = Dmatrix(D,Qbar2,z2,z3)

D =

    7.9983    2.7832    3.8604
    2.7832    1.6983    1.1759
    1.9302    0.5879    5.4232

>> D = Dmatrix(D,Qbar3,z3,z4)

D =

    8.5240    2.7933    3.8604
    2.7933    1.7394    1.1759
    1.9302    0.5879    5.4381

>> D = Dmatrix(D,Qbar4,z4,z5)

D =

    9.0496    2.8035    3.8604
    2.8035    1.7804    1.1759
    1.9302    0.5879    5.4529
```

```
>> D = Dmatrix(D,Qbar5,z5,z6)

D =

   11.2030    3.5528    1.6085
    3.5528    2.2376    0.4900
    0.8042    0.2450    6.9130

>> D = Dmatrix(D,Qbar6,z6,z7)

D =

   17.0479    5.5867    7.7207
    5.5867    3.4787    2.3518
    3.8604    1.1759   10.8762

>> D = D/3

D =

    5.6826    1.8622    2.5736
    1.8622    1.1596    0.7839
    1.2868    0.3920    3.6254
```

Problems

Problem 8.1

Derive (8.5) in detail.

MATLAB Problem 8.2

Consider a graphite-reinforced polymer composite laminate with the elastic constants as given in Example 2.2. The laminate has total thickness of 0.600 mm and is stacked as a $[0/90]_S$ laminate. The four layers are of equal thickness. Calculate the $[A]$, $[B]$, and $[D]$ matrices for this laminate.

MATLAB Problem 8.3

Consider a glass-reinforced polymer composite laminate with the elastic constants as given in Problem 2.7. The laminate has total thickness of 0.600 mm and is stacked as a $[0/90]_S$ laminate. The four layers are of equal thickness. Calculate the $[A]$, $[B]$, and $[D]$ matrices for this laminate.

MATLAB Problem 8.4

Consider a glass-reinforced polymer composite laminate with the elastic constants as given in Problem 2.7. The laminate has total thickness of 0.900 mm and is stacked as a $[\pm 30/0]_S$ laminate. The six layers are of equal thickness. Calculate the $[A]$, $[B]$, and $[D]$ matrices for this laminate.

MATLAB Problem 8.5

Consider a graphite-reinforced polymer composite laminate with the elastic constants as given in Example 2.2. The laminate has total thickness of 0.800 mm and is stacked as a $[+30/0]_S$ laminate. The four layers are of equal thickness. Calculate the $[A]$, $[B]$, and $[D]$ matrices for this laminate.

MATLAB Problem 8.6

Consider a glass-reinforced polymer composite laminate with the elastic constants as given in Problem 2.7. The laminate has total thickness of 0.800 mm and is stacked as a $[+30/0]_S$ laminate. The four layers are of equal thickness. Calculate the $[A]$, $[B]$, and $[D]$ matrices for this laminate.

MATLAB Problem 8.7

Consider a graphite-reinforced polymer composite laminate with the elastic constants as given in Example 2.2. The laminate has total thickness of 0.600 mm and is stacked as a $[+45/0/-30]_T$ laminate. The three layers are of equal thickness. Calculate the $[A]$, $[B]$, and $[D]$ matrices for this laminate.

MATLAB Problem 8.8

Consider a glass-reinforced polymer composite laminate with the elastic constants as given in Problem 2.7. The laminate has total thickness of 0.600 mm and is stacked as a $[+45/0/-30]_T$ laminate. The three layers are of equal thickness. Calculate the $[A]$, $[B]$, and $[D]$ matrices for this laminate.

9

Effective Elastic Constants of a Laminate

9.1 Basic Equations

In this chapter, we introduce the concept of *effective elastic constants* for the laminate. These constants are the effective extensional modulus in the x direction $\bar{E}_x$, the effective extensional modulus in the y direction $\bar{E}_y$, the effective Poisson's ratios $\bar{\nu}_{xy}$ and $\bar{\nu}_{yx}$, and the effective shear modulus in the x-y plane $\bar{G}_{xy}$.

The effective elastic constants are usually defined when considering the inplane loading of symmetric balanced laminates. In the following equations, we consider only symmetric balanced or symmetric cross-ply laminates. We therefore define the following three average laminate stresses [1]:

$$\bar{\sigma}_x = \frac{1}{H} \int_{-H/2}^{H/2} \sigma_x dz \tag{9.1}$$

$$\bar{\sigma}_y = \frac{1}{H} \int_{-H/2}^{H/2} \sigma_y dz \tag{9.2}$$

$$\bar{\tau}_{xy} = \frac{1}{H} \int_{-H/2}^{H/2} \tau_{xy} dz \tag{9.3}$$

where H is the thickness of the laminate. Comparing (9.1), (9.2), and (9.3) with (7.13), we obtain the following relations between the average stresses and the force resultants:

$$\bar{\sigma}_x = \frac{1}{H} N_x \tag{9.4}$$

$$\bar{\sigma}_y = \frac{1}{H} N_y \tag{9.5}$$

$$\bar{\tau}_{xy} = \frac{1}{H} N_{xy} \tag{9.6}$$

Solving (9.4), (9.5), and (9.6) for N_x, N_y, and N_{xy}, and substituting the results into (8.11) and (8.12) for symmetric balanced laminates, we obtain:

$$\begin{Bmatrix} \varepsilon_x^0 \\ \varepsilon_y^0 \\ \gamma_{xy}^0 \end{Bmatrix} = \begin{bmatrix} a_{11}H & a_{12}H & 0 \\ a_{12}H & a_{22}H & 0 \\ 0 & 0 & a_{66}H \end{bmatrix} \begin{Bmatrix} \bar{\sigma}_x \\ \bar{\sigma}_y \\ \bar{\tau}_{xy} \end{Bmatrix} \tag{9.7}$$

The above 3 × 3 matrix is defined as the *laminate compliance matrix* for symmetric balanced laminates. Therefore, by analogy with (4.5), we obtain the following effective elastic constants for the laminate:

$$\bar{E}_x = \frac{1}{a_{11}H} \tag{9.8a}$$

$$\bar{E}_y = \frac{1}{a_{22}H} \tag{9.8b}$$

$$\bar{G}_{xy} = \frac{1}{a_{66}H} \tag{9.8c}$$

$$\bar{\nu}_{xy} = -\frac{a_{12}}{a_{11}} \tag{9.8d}$$

$$\bar{\nu}_{yx} = -\frac{a_{12}}{a_{22}} \tag{9.8e}$$

It is clear from the above equations that $\bar{\nu}_{xy}$ and $\bar{\nu}_{yx}$ are not independent and are related by the following reciprocity relation:

$$\frac{\bar{\nu}_{xy}}{\bar{E}_x} = \frac{\bar{\nu}_{yx}}{\bar{E}_y} \tag{9.9}$$

Finally, we note that the expressions of the effective elastic constants of (9.8) can be re-written in terms of the components A_{ij} of the matrix $[A]$ as shown in Example 9.1.

9.2 MATLAB Functions Used

The five MATLAB function used in this chapter to calculate the average laminate elastic constants are:

Ebarx(A, H) – This function calculates the average laminate modulus in the x-direction $\bar{E}_x$. There are two input arguments to this function – they are the thickness of the laminate H and the 3 × 3 stiffness matrix $[A]$ for balanced symmetric laminates. The function returns a scalar quantity which the desired modulus.

Ebary(A, H) – This function calculates the average laminate modulus in the y-direction $\bar{E}_y$. There are two input arguments to this function – they are the thickness of the laminate H and the 3 × 3 stiffness matrix $[A]$ for balanced symmetric laminates. The function returns a scalar quantity which the desired modulus.

NUbarxy(A, H) – This function calculates the average laminate Poisson's ratio $\bar{\nu}_{xy}$. There are two input arguments to this function – they are the thickness of the laminate H and the 3 × 3 stiffness matrix $[A]$ for balanced symmetric laminates. The function returns a scalar quantity which the desired Poisson's ratio.

NUbaryx(A, H) – This function calculates the average laminate Poisson's ratio $\bar{\nu}_{yx}$. There are two input arguments to this function – they are the thickness of the laminate H and the 3×3 stiffness matrix $[A]$ for balanced symmetric laminates. The function returns a scalar quantity which the desired Poisson's ratio.

Gbarxy(A, H) – This function calculates the average laminate shear modulus $\bar{G}_{xy}$. There are two input arguments to this function – they are the thickness of the laminate H and the 3×3 stiffness matrix $[A]$ for balanced symmetric laminates. The function returns a scalar quantity which the desired shear modulus.

The following is a listing of the MATLAB source code for these functions:

```
function y = Ebarx(A,H)
%Ebarx   This function returns the average laminate modulus
%        in the x-direction. Its input are two arguments:
%        A  - 3 x 3 stiffness matrix for balanced symmetric
%             laminates.
%        H - thickness of laminate
a = inv(A);
y = 1/(H*a(1,1));
```

```
function y = Ebary(A,H)
%Ebary   This function returns the average laminate modulus
%        in the y-direction. Its input are two arguments:
%        A  - 3 x 3 stiffness matrix for balanced symmetric
%             laminates.
%        H - thickness of laminate
a = inv(A);
y = 1/(H*a(2,2));
```

```
function y = NUbarxy(A,H)
%NUbarxy   This function returns the average laminate
%          Poisson's ratio NUxy. Its input are two arguments:
%          A  - 3 x 3 stiffness matrix for balanced symmetric
%               laminates.
%          H - thickness of laminate
a = inv(A);
y = -a(1,2)/a(1,1);
```

```
function y = NUbaryx(A,H)
%NUbaryx   This function returns the average laminate
%          Poisson's ratio NUyx. Its input are two arguments:
%          A  - 3 x 3 stiffness matrix for balanced symmetric
%               laminates.
%          H - thickness of laminate
a = inv(A);
y = -a(1,2)/a(2,2);
```

```
function y = Gbarxy(A,H)
%Gbarxy   This function returns the average laminate shear
%         modulus. Its input are two arguments:
%        A  - 3 x 3 stiffness matrix for balanced symmetric
```

```
%             laminates.
%        H - thickness of laminate
a = inv(A);
y = 1/(H*a(3,3));
```

Example 9.1

Show that the effective elastic constants for the laminate can be written in terms of the components A_{ij} of the $[A]$ matrix as follows:

$$\bar{E}_x = \frac{A_{11}AA_{22} - A_{12}^2}{A_{22}H} \tag{9.10a}$$

$$\bar{E}_y = \frac{A_{11}AA_{22} - A_{12}^2}{A_{11}H} \tag{9.10b}$$

$$\bar{\nu}_{xy} = \frac{A_{12}}{A_{22}} \tag{9.10c}$$

$$\bar{\nu}_{yx} = \frac{A_{12}}{A_{11}} \tag{9.10d}$$

$$\bar{G}_{xy} = \frac{A_{66}}{H} \tag{9.10e}$$

Solution

Starting with (8.11) and (8.12) as follows:

$$\begin{Bmatrix} N_x \\ N_y \\ N_{xy} \end{Bmatrix} = \begin{bmatrix} A_{11} & A_{12} & 0 \\ A_{12} & A_{22} & 0 \\ 0 & 0 & A_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_x^0 \\ \varepsilon_y^0 \\ \gamma_{xy}^0 \end{Bmatrix} \tag{9.11}$$

take the inverse of (9.11) to obtain:

$$\begin{Bmatrix} \varepsilon_x^0 \\ \varepsilon_y^0 \\ \gamma_{xy}^0 \end{Bmatrix} = \begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{12} & a_{22} & 0 \\ 0 & 0 & a_{66} \end{bmatrix} \begin{Bmatrix} N_x \\ N_y \\ N_{xy} \end{Bmatrix} \tag{9.12}$$

where

$$a_{11} = \frac{A_{22}}{A_{11}A_{22} - A_{12}^2} \tag{9.13a}$$

$$a_{22} = \frac{A_{11}}{A_{11}A_{22} - A_{12}^2} \tag{9.13b}$$

$$a_{12} = \frac{A_{12}}{A_{11}A_{22} - A_{12}^2} \tag{9.13c}$$

$$a_{66} = \frac{1}{A_{66}} \tag{9.13d}$$

Next, substitute (9.13) into (9.8) to obtain the required expressions as follows:

$$\bar{E}_x = \frac{A_{11}AA_{22} - A_{12}^2}{A_{22}H} \tag{9.14a}$$

$$\bar{E}_y = \frac{A_{11}AA_{22} - A_{12}^2}{A_{11}H} \tag{9.14b}$$

$$\bar{\nu}_{xy} = \frac{A_{12}}{A_{22}} \tag{9.14c}$$

$$\bar{\nu}_{yx} = \frac{A_{12}}{A_{11}} \tag{9.14d}$$

$$\bar{G}_{xy} = \frac{A_{66}}{H} \tag{9.14e}$$

MATLAB Example 9.2

Consider a four-layer $[0/90]_S$ graphite-reinforced polymer composite laminate with the elastic constants as given in Example 2.2. The laminate has total thickness of 0.800 mm. The four layers are of equal thickness. Use MATLAB to determine the five effective elastic constants for this laminate.

Solution

This example is solved using MATLAB. First, the reduced stiffness matrix $[Q]$ for a typical layer using the MATLAB function *ReducedStiffness* as follows:

```
EDU>> Q = ReducedStiffness(155.0, 12.10, 0.248, 4.40)

Q =

  155.7478    3.0153         0
    3.0153   12.1584         0
         0         0    4.4000
```

Next, the transformed reduced stiffness matrix $\left[\bar{Q}\right]$ is calculated for each layer using the MATLAB function *Qbar* as follows:

```
EDU>> Qbar1 = Qbar(Q, 0)

Qbar1 =

  155.7478    3.0153         0
    3.0153   12.1584         0
         0         0    4.4000
```

```
EDU>> Qbar2 = Qbar(Q, 90)

Qbar2 =

   12.1584    3.0153   -0.0000
    3.0153  155.7478    0.0000
   -0.0000    0.0000    4.4000

EDU>> Qbar3 = Qbar(Q, 90)

Qbar3 =

   12.1584    3.0153   -0.0000
    3.0153  155.7478    0.0000
   -0.0000    0.0000    4.4000

EDU>> Qbar4 = Qbar(Q, 0)

Qbar4 =

  155.7478    3.0153         0
    3.0153   12.1584         0
         0         0    4.4000
```

Next, the distances $z_k (k = 1, 2, 3, 4, 5)$ are calculated as follows:

```
EDU>> z1 = -0.400

z1 =

   -0.4000

EDU>> z2 = -0.200

z2 =

   -0.2000

EDU>> z3 = 0

z3 =

     0

EDU>> z4 = 0.200

z4 =

    0.2000
```

```
EDU>> z5 = 0.400

z5 =

    0.4000
```

Next, the $[A]$ matrix is calculated using four calls to the MATLAB function *Amatrix* as follows:

```
EDU>> A = zeros(3,3)

A =

     0     0     0
     0     0     0
     0     0     0

EDU>> A = Amatrix(A,Qbar1,z1,z2)

A =

   31.1496    0.6031         0
    0.6031    2.4317         0
         0         0    0.8800

EDU>> A = Amatrix(A,Qbar2,z2,z3)

A =

   33.5812    1.2061   -0.0000
    1.2061   33.5812    0.0000
   -0.0000    0.0000    1.7600

EDU>> A = Amatrix(A,Qbar3,z3,z4)

A =

   36.0129    1.8092   -0.0000
    1.8092   64.7308    0.0000
   -0.0000    0.0000    2.6400

EDU>> A = Amatrix(A,Qbar4,z4,z5)

A =

   67.1625    2.4122   -0.0000
    2.4122   67.1625    0.0000
   -0.0000    0.0000    3.5200
```

Finally, five calls are made to the five MATLAB functions introduced in this chapter to calculate the five effective elastic constants of this laminate.

```
EDU>> H = 0.800
H =

    0.8000

EDU>> Ebarx(A,H)

ans =

   83.8448

EDU>> Ebary(A,H)

ans =

   83.8448

EDU>> NUbarxy(A,H)

ans =

    0.0359

EDU>> NUbaryx(A,H)

ans =

    0.0359

EDU>> Gbarxy(A,H)

ans =

    4.4000
```

MATLAB Example 9.3

Consider a six-layer $[\pm 30/0]_S$ graphite-reinforced polymer composite laminate with the elastic constants as given in Example 2.2. The laminate has total thickness of 0.900 mm. The four layers are of equal thickness. Use MATLAB to determine the five effective elastic constants for this laminate.

Solution

This example is solved using MATLAB. First, the reduced stiffness matrix $[Q]$ for a typical layer using the MATLAB function *ReducedStiffness* as follows:

```
EDU>> Q = ReducedStiffness(155.0, 12.10, 0.248, 4.40)

Q =

  155.7478    3.0153         0
    3.0153   12.1584         0
         0         0    4.4000
```

Next, the transformed reduced stiffness matrix $[\bar{Q}]$ is calculated for each layer using the MATLAB function *Qbar* as follows:

```
EDU>> Qbar1 = Qbar(Q, 30)

Qbar1 =

   91.1488   31.7170   95.3179
   31.7170   19.3541   29.0342
   47.6589   14.5171   61.8034

EDU>> Qbar2 = Qbar(Q, -30)

Qbar2 =

   91.1488   31.7170  -95.3179
   31.7170   19.3541  -29.0342
  -47.6589  -14.5171   61.8034

EDU>> Qbar3 = Qbar(Q, 0)

Qbar3 =

  155.7478    3.0153         0
    3.0153   12.1584         0
         0         0    4.4000

EDU>> Qbar4 = Qbar(Q, 0)

Qbar4 =

  155.7478    3.0153         0
    3.0153   12.1584         0
         0         0    4.4000

EDU>> Qbar5 = Qbar(Q, -30)

Qbar5 =

   91.1488   31.7170  -95.3179
   31.7170   19.3541  -29.0342
  -47.6589  -14.5171   61.8034
```

```
EDU>> Qbar6 = Qbar(Q, 30)

Qbar6 =

   91.1488   31.7170   95.3179
   31.7170   19.3541   29.0342
   47.6589   14.5171   61.8034
```

Next, the distances z_k ($k = 1, 2, 3, 4, 5, 6, 7$) are calculated as follows:

```
EDU>> z1 = -0.450

z1 =

   -0.4500

EDU>> z2 = -0.300

z2 =

   -0.3000

EDU>> z3 = -0.150

z3 =

   -0.1500

EDU>> z4 = 0

z4 =

     0

EDU>> z5 = 0.150

z5 =

    0.1500

EDU>> z6 = 0.300

z6 =

    0.3000

EDU>> z7 = 0.450

z7 =
    0.4500
```

Next, the $[A]$ matrix is calculated using six calls to the MATLAB function *Amatrix* as follows:

```
EDU>> A = zeros(3,3)

A =

     0     0     0
     0     0     0
     0     0     0

EDU>> A = Amatrix(A,Qbar1,z1,z2)

A =

   13.6723    4.7575   14.2977
    4.7575    2.9031    4.3551
    7.1488    2.1776    9.2705

EDU>> A = Amatrix(A,Qbar2,z2,z3)

A =

   27.3446    9.5151    0.0000
    9.5151    5.8062    0.0000
    0.0000    0.0000   18.5410

EDU>> A = Amatrix(A,Qbar3,z3,z4)

A =

   50.7068    9.9674    0.0000
    9.9674    7.6300    0.0000
    0.0000    0.0000   19.2010

EDU>> A = Amatrix(A,Qbar4,z4,z5)

A =

   74.0690   10.4197    0.0000
   10.4197    9.4537    0.0000
    0.0000    0.0000   19.8610

EDU>> A = Amatrix(A,Qbar5,z5,z6)

A =
   87.7413   15.1772  -14.2977
   15.1772   12.3568   -4.3551
   -7.1488   -2.1776   29.1315
```

```
EDU>> A = Amatrix(A,Qbar6,z6,z7)

A =

  101.4136   19.9348    0.0000
   19.9348   15.2599    0.0000
    0.0000    0.0000   38.4020
```

Finally, five calls are made to the five MATLAB functions introduced in this chapter to calculate the five effective elastic constants of this laminate.

```
EDU>> H = 0.900

H =

    0.9000

EDU>> Ebarx(A,H)

ans =

   83.7466

EDU>> Ebary(A,H)

ans =

   12.6015

EDU>> NUbarxy(A,H)

ans =

    1.3063

EDU>> NUbaryx(A,H)

ans =

    0.1966

EDU>> Gbarxy(A,H)

ans =

   42.6689
```

Problems

Problem 9.1

Show that the effective shear modulus $\bar{G}_{xy}$ is *not* related to the effective extensional modulus $\bar{E}_x$ and the effective Poisson's ratio $\bar{\nu}_{xy}$ by the relation:

$$\bar{G}_{xy} = \frac{\bar{E}_x}{2(1+\bar{\nu}_{xy})} \tag{9.15}$$

MATLAB Problem 9.2

Consider a graphite-reinforced polymer composite laminate with the elastic constants as given in Example 2.2. The laminate has total thickness of 0.600 mm and is stacked as a $[0/90]_S$ laminate. The four layers are of equal thickness. Use MATLAB to determine the five effective elastic constants for this laminate.

MATLAB Problem 9.3

Consider a glass-reinforced polymer composite laminate with the elastic constants as given in Problem 2.7. The laminate has total thickness of 0.600 mm and is stacked as a $[0/90]_S$ laminate. The four layers are of equal thickness. Use MATLAB to determine the five effective elastic constants for this laminate.

MATLAB Problem 9.4

Consider a glass-reinforced polymer composite laminate with the elastic constants as given in Problem 2.7. The laminate has total thickness of 0.900 mm and is stacked as a $[\pm 30/0]_S$ laminate. The six layers are of equal thickness. Use MATLAB to determine the five effective elastic constants for this laminate.

MATLAB Problem 9.5

Consider a graphite-reinforced polymer composite laminate with the elastic constants as given in Example 2.2. The laminate has total thickness of 0.800 mm and is stacked as a $[+30/0]_S$ laminate. The four layers are of equal thickness. Use MATLAB to determine the five effective elastic constants for this laminate.

MATLAB Problem 9.6

Consider a glass-reinforced polymer composite laminate with the elastic constants as given in Problem 2.7. The laminate has total thickness of 0.800 mm and is stacked as a $[+30/0]_S$ laminate. The four layers are of equal thickness. Use MATLAB to determine the five effective elastic constants for this laminate.

MATLAB Problem 9.7

Consider a graphite-reinforced polymer composite laminate with the elastic constants as given in Example 2.2. The laminate has total thickness of 0.600 mm and is stacked as a $[+45/0/-30]_T$ laminate. The three layers are of equal thickness. Use MATLAB to determine the five effective elastic constants for this laminate.

MATLAB Problem 9.8

Consider a glass-reinforced polymer composite laminate with the elastic constants as given in Problem 2.7. The laminate has total thickness of 0.600 mm and is stacked as a $[+45/0/-30]_T$ laminate. The three layers are of equal thickness. Use MATLAB to determine the five effective elastic constants for this laminate.

10

Failure Theories of a Lamina

10.1 Basic Equations

In this chapter we present various failure theories for one single layer of the composite laminate, usually called a lamina. We use the following notation throughout this chapter for the various strengths or ultimate stresses:

σ_1^T : tensile strength in longitudinal direction.
σ_1^C : compressive strength in longitudinal direction.
σ_2^T : tensile strength in transverse direction.
σ_2^C : compressive strength in transverse direction.
τ_{12}^F : shear strength in the 1-2 plane.

where the strength means the ultimate stress or failure stress, the longitudinal direction is the fiber direction (1-direction), and the transverse direction is the 2-direction (perpendicular to the fiber).

We also use the following notation for the ultimate strains:

ε_1^T : ultimate tensile strain in the longitudinal direction.
ε_1^C : ultimate compressive strain in the longitudinal direction.
ε_2^T : ultimate tensile strain in the transverse direction.
ε_2^C : ultimate compressive strain in the transverse direction.
γ_{12}^F : ultimate shear strain in the 1-2 plane.

It is assumed that the lamina behaves in a linear elastic manner. For the longitudinal uniaxial loading of the lamina (see Fig. 10.1), we have the following elastic relations:

$$\sigma_1^T = E_1 \varepsilon_1^T \tag{10.1}$$
$$\sigma_1^C = E_1 \varepsilon_1^C \tag{10.2}$$

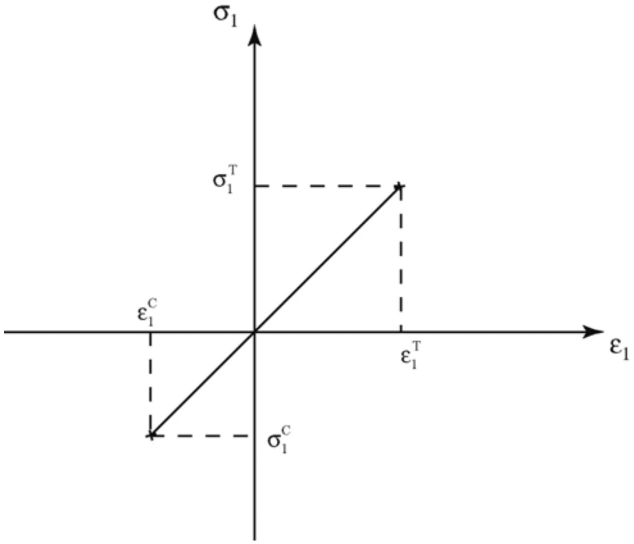

Fig. 10.1. Stress-strain curve for the longitudinal uniaxial loading of a lamina

where E_1 is Young's modulus of the lamina in the longitudinal (fiber) direction.

For the transverse uniaxial loading of the lamina (see Fig. 10.2), we have the following elastic relations:

$$\sigma_2^T = E_2 \varepsilon_2^T \tag{10.3}$$

$$\sigma_2^C = E_2 \varepsilon_2^C \tag{10.4}$$

where E_2 is Young's modulus of the lamina in the transverse direction. For the shear loading of the lamina (see Fig. 10.3), we have the following elastic relation:

$$\tau_{12}^F = G_{12} \gamma_{12}^F \tag{10.5}$$

where G_{12} is the shear modulus of the lamina.

10.1.1 Maximum Stress Failure Theory

In the *maximum stress failure theory*, failure of the lamina is assumed to occur whenever any normal or shear stress component equals or exceeds the corresponding strength. This theory is written mathematically as follows:

$$\sigma_1^C < \sigma_1 < \sigma_1^T \tag{10.6}$$

$$\sigma_2^C < \sigma_2 < \sigma_2^T \tag{10.7}$$

$$|\tau_{12}| < \tau_{12}^F \tag{10.8}$$

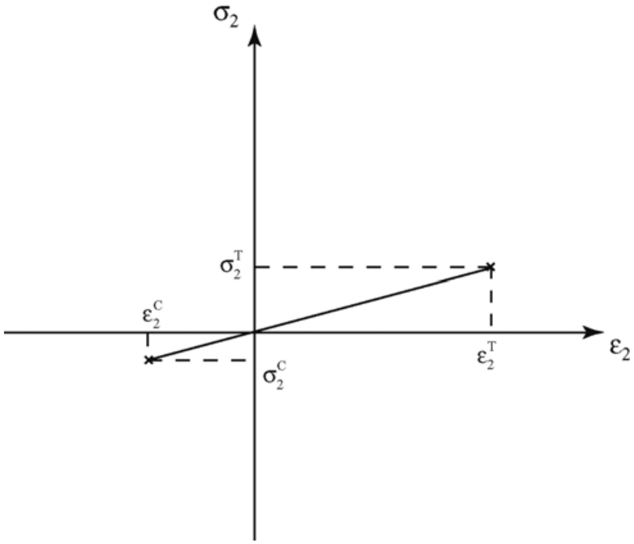

Fig. 10.2. Stress-strain curve for the transverse uniaxial loading of a lamina

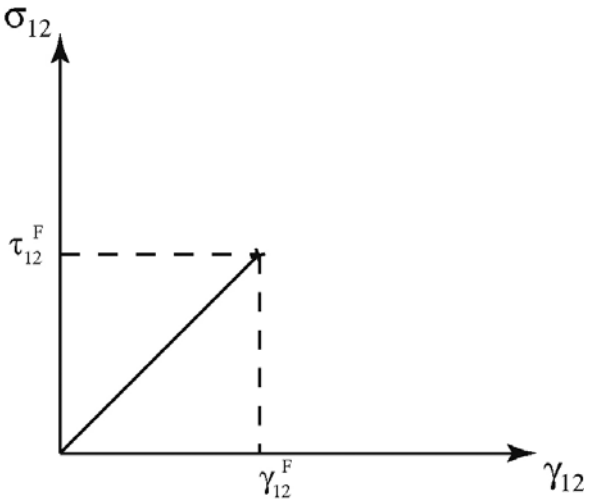

Fig. 10.3. Stress-strain curve for the shear loading of a lamina

where σ_1 and σ_2 are the maximum material normal stresses in the lamina, while τ_{12} is the maximum shear stress in the lamina.

The failure envelope for this theory is clearly illustrated in Fig. 10.4. The advantage of this theory is that it is simple to use but the major disadvantage is that there is no interaction between the stress components.

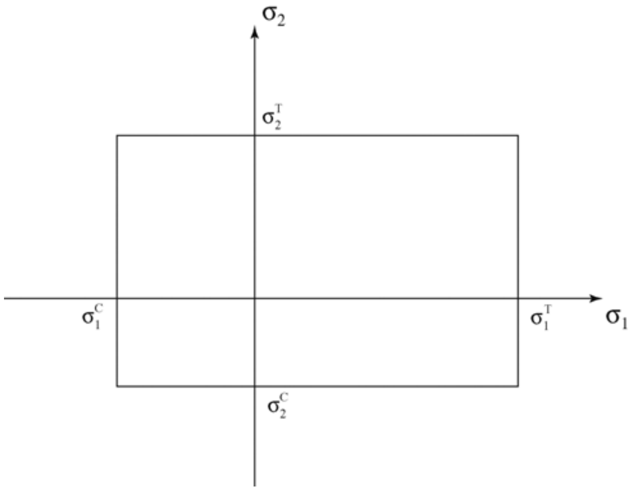

Fig. 10.4. Failure envelope for the maximum stress failure theory

10.1.2 Maximum Strain Failure Theory

In the *maximum strain failure theory*, failure of the lamina is assumed to occur whenever any normal or shear strain component equals or exceeds the corresponding ultimate strain. This theory is written mathematically as follows:

$$\varepsilon_1^C < \varepsilon_1 < \varepsilon_1^T \tag{10.9}$$

$$\varepsilon_2^C < \varepsilon_2 < \varepsilon_2^T \tag{10.10}$$

$$|\gamma_{12}| < \gamma_{12}^F \tag{10.11}$$

where ε_1, ε_2, and γ_{12} are the principal material axis strain components. In this case, we have the following relation between the strains and the stresses in the longitudinal direction:

$$\varepsilon_1 = \frac{\sigma_1^T}{E_1} = \frac{\sigma_1}{E_1} - \nu_{12}\frac{\sigma_2}{E_1} \tag{10.12}$$

Simplifying (10.12), we obtain:

$$\sigma_2 = \frac{\sigma_1 - \sigma_1^T}{\nu_{12}} \tag{10.13}$$

Similarly, we have the following relation between the strains and the stresses in the transverse direction:

$$\varepsilon_2 = \frac{\sigma_2^T}{E_2} = \frac{\sigma_2}{E_2} - \nu_{21}\frac{\sigma_1}{E_2} \tag{10.14}$$

Simplifying (10.14), we obtain:

$$\sigma_2 = \nu_{21}\sigma_1 + \sigma_2^T \tag{10.15}$$

The failure envelope for this theory is clearly shown in Fig. 10.5 (based on (10.13) and (10.15)). The advantage of this theory is that it is simple to use but the major disadvantage is that there is no interaction between the strain components.

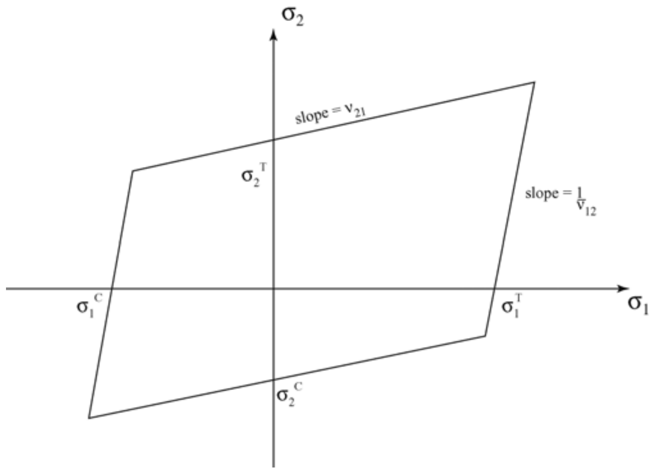

Fig. 10.5. Failure envelope for the maximum strain failure theory

Figure 10.6 shows the two failure envelopes of the maximum stress theory and the maximum strain theory superimposed on the same plot for comparison.

10.1.3 Tsai-Hill Failure Theory

The *Tsai-Hill failure theory* is derived from the von Mises distortional energy yield criterion for isotropic materials but is applied to anisotropic materials with the appropriate modifications. In this theory, failure is assumed to occur whenever the distortional yield energy equals or exceeds a certain value related to the strength of the lamina. In this theory, there is no distinction between the tensile and compressive strengths. Therefore, we use the following notation for the strengths of the lamina:

σ_1^F : strength in longitudinal direction.
σ_2^F : strength in transverse direction.
τ_{12}^F : shear strength in the 1-2 plane.

The Tsai-Hill failure theory is written mathematically for the lamina as follows:

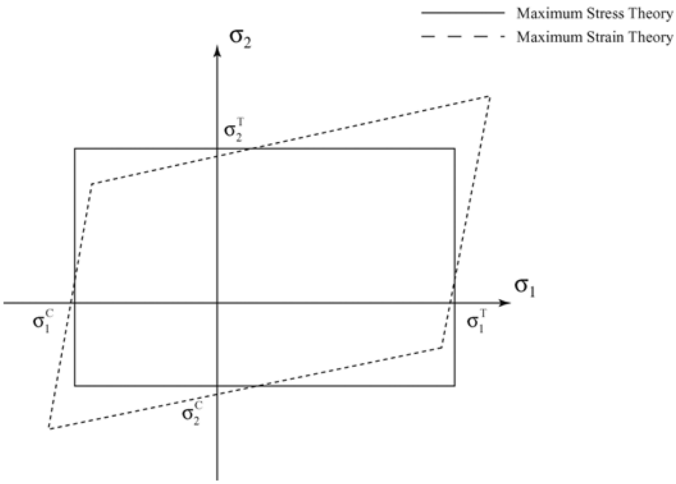

Fig. 10.6. Comparison of the failure envelopes for the maximum stress theory and maximum strain theory

$$\frac{\sigma_1^2}{\left(\sigma_1^F\right)^2} - \frac{\sigma_1\sigma_2}{\left(\sigma_1^F\right)^2} + \frac{\sigma_2^2}{\left(\sigma_2^F\right)^2} + \frac{\tau_{12}^2}{\left(\tau_{12}^F\right)^2} \leq 1 \tag{10.16}$$

The failure envelope for this theory is clearly shown in Fig. 10.7. The advantage of this theory is that there is interaction between the stress components. However, this theory does not distinguish between the tensile and compressive strengths and is not as simple to use as the maximum stress theory or the maximum strain theory.

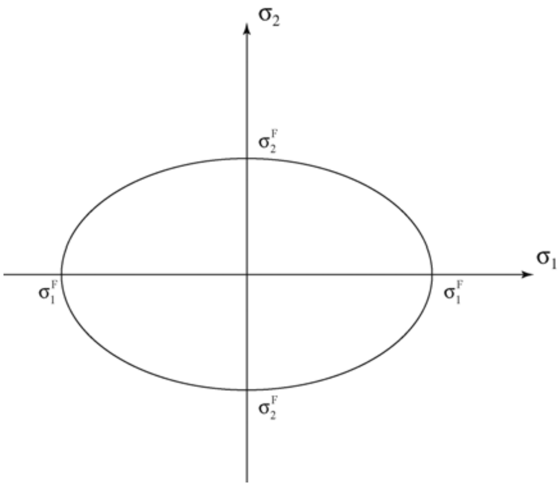

Fig. 10.7. Failure envelope for the Tsai-Hill failure theory

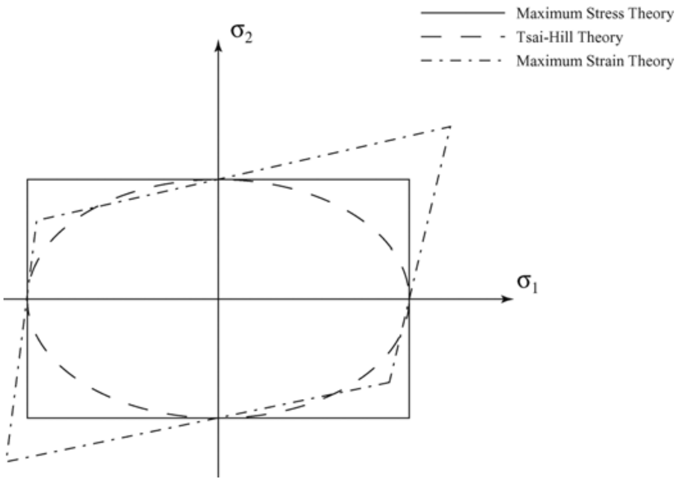

Fig. 10.8. Comparison between the three failure envelopes

Figure 10.8 shows the three failure envelopes of the maximum stress theory, the maximum strain theory, and the Tsai-Hill theory superimposed on the same plot for comparison.

10.1.4 Tsai-Wu Failure Theory

The *Tsai-Wu failure theory* is based on a total strain energy failure theory. In this theory, failure is assumed to occur in the lamina if the following condition is satisfied:

$$F_{11}\sigma_1^2 + F_{22}\sigma_2^2 + F_{66}\tau_{12}^2 + F_1\sigma_1 + F_2\sigma_2 + F_{12}\sigma_1\sigma_2 \leq 1 \tag{10.17}$$

where the coefficients F_{11}, F_{22}, F_{66}, F_1, F_2, and F_{12} are given by:

$$F_{11} = \frac{1}{\sigma_1^T \sigma_1^C} \tag{10.18}$$

$$F_{22} = \frac{1}{\sigma_2^T \sigma_2^C} \tag{10.19}$$

$$F_1 = \frac{1}{\sigma_1^T} - \frac{1}{\sigma_1^C} \tag{10.20}$$

$$F_1 = \frac{1}{\sigma_2^T} - \frac{1}{\sigma_2^C} \tag{10.21}$$

$$F_{66} = \frac{1}{\left(\tau_{12}^F\right)^2} \tag{10.22}$$

and F_{12} is a coefficient that is determined experimentally. Tsai-Hahn determined F_{12} to be given by the following approximate expression:

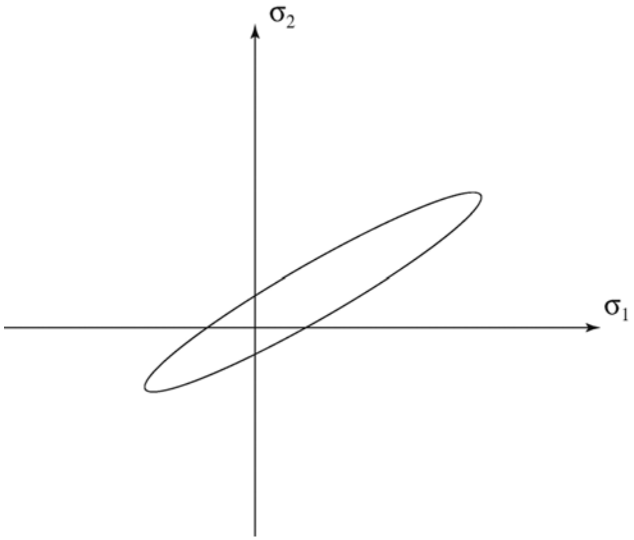

Fig. 10.9. A general failure ellipse for the Tsai-Wu failure theory

$$F_{12} \approx -\frac{1}{2}\sqrt{F_{11}F_{22}} \tag{10.23}$$

The failure envelope for this theory is shown in general in Fig. 10.9. The advantage of this theory is that there is interaction between the stress components and the theory does distinguish between the tensile and compressive strengths. A major disadvantage of this theory is that it is not simple to use.

Finally, in order to compare the failure envelopes for a composite lamina with the envelopes of isotropic ductile materials, Fig. 10.10 shows the failure envelopes for the usual von Mises and Tresca criteria for isotropic materials.

Problems

Problem 10.1

Determine the maximum value of $\alpha > 0$ if stresses of $\sigma_x = 3\alpha$, $\sigma_y = -2\alpha$, and $\tau_{xy} = 5\alpha$ are applied to a 60°-lamina of graphite/epoxy. Use the maximum stress failure theory. The material properties of this lamina are given as follows:

$$\begin{aligned}
V^f &= 0.70 & \sigma_1^T &= 1500\,\mathrm{MPa} \\
E_1 &= 181\,\mathrm{GPa} & \sigma_1^C &= 1500\,\mathrm{MPa} \\
E_2 &= 10.30\,\mathrm{GPa} & \sigma_2^T &= 40\,\mathrm{MPa} \\
\nu_{12} &= 0.28 & \sigma_2^C &= 246\,\mathrm{MPa} \\
G_{12} &= 7.17\,\mathrm{GPa} & \tau_{12}^F &= 68\,\mathrm{MPa}
\end{aligned}$$

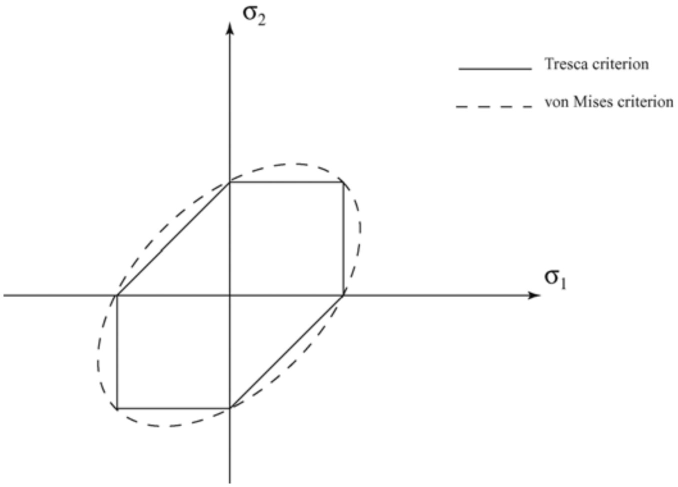

Fig. 10.10. Two failure criteria for ductile homogeneous materials

Problem 10.2

Repeat Problem 10.1 using the maximum strain failure theory instead of the maximum stress failure theory.

Problem 10.3

Repeat Problem 10.1 using the Tsai-Hill failure theory instead of the maximum stress failure theory.

Problem 10.4

Repeat Problem 10.1 using the Tsai-Wu failure theory instead of the maximum stress failure theory.

MATLAB Problem 10.5

Use MATLAB to plot the four failure envelopes using the strengths given in Problem 10.1.

Problem 10.6

Determine the maximum value of $\alpha > 0$ if stresses of $\sigma_x = 3\alpha$, $\sigma_y = -2\alpha$, and $\tau_{xy} = 5\alpha$ are applied to a 30°-lamina of glass/epoxy. Use the maximum stress failure theory. The material properties of this lamina are given as follows:

$$
\begin{aligned}
V^f &= 0.45 & \sigma_1^T &= 1062\,\text{MPa} \\
E_1 &= 38.6\,\text{GPa} & \sigma_1^C &= 610\,\text{MPa} \\
E_2 &= 8.27\,\text{GPa} & \sigma_2^T &= 31\,\text{MPa} \\
\nu_{12} &= 0.26 & \sigma_2^C &= 118\,\text{MPa} \\
G_{12} &= 4.14\,\text{GPa} & \tau_{12}^F &= 72\,\text{MPa}
\end{aligned}
$$

Problem 10.7

Repeat Problem 10.6 using the maximum strain failure theory instead of the maximum stress failure theory.

Problem 10.8

Repeat Problem 10.6 using the Tsai-Hill failure theory instead of the maximum stress failure theory.

Problem 10.9

Repeat Problem 10.6 using the Tsai-Wu failure theory instead of the maximum stress failure theory.

MATLAB Problem 10.10

Use MATLAB to plot the four failure envelopes using the strengths given in Problem 10.6.

11

Introduction to Homogenization of Composite Materials

11.1 Eshelby Method

In this chapter, we present a brief overview of the homogenization of composite materials. Homogenization refers to the process of considering a statistically homogeneous representation of the composite material called a *representative volume element (RVE)*. This homogenized element is considered for purposes of calculating the stresses and strains in the matrix and fibers. We will emphasize mainly the *Eshelby method* in the homogenization process. For more details, the reader is referred to the book *An Introduction to Metal Matrix Composites* by Clyne and Withers.

Since the composite system is composed of two different materials (matrix and fibers) with two different stiffnesses, internal stresses will arise in both the two constituents. Eshelby in the 1950s demonstrated that an analytical solution may be obtained for the special case when the fibers have the shape of an ellipsoid. Furthermore, the stress is assumed to be uniform within the ellipsoid. Eshelby's method is summarized by representing the actual inclusion (i.e fibers) by one made of the matrix material (called the *equivalent homogeneous inclusion*). This equivalent inclusion is assumed to have an appropriate strain (called the *equivalent transformation strain*) such that the stress field is the same as for the actual inclusion. This is the essence of the homogenization process.

The following is a summary of the steps followed in the homogenization procedure according to the Eshelby method (see Fig. 11.1):

1. Consider an initially unstressed elastic homogeneous material (see Fig. 11.1a). Imagine cutting an ellipsoidal region (i.e. inclusion) from this material. Imagine also that the inclusion undergoes a shape change free from the constraining matrix by subjecting it to a transformation strain ε_{ij}^T (see Fig. 11.1b) where the indices i and j take the values 1, 2, and 3.
2. Since the inclusion has now changed in shape, it cannot be replaced directly into the hole in the matrix material. Imagine applying surface tractions to

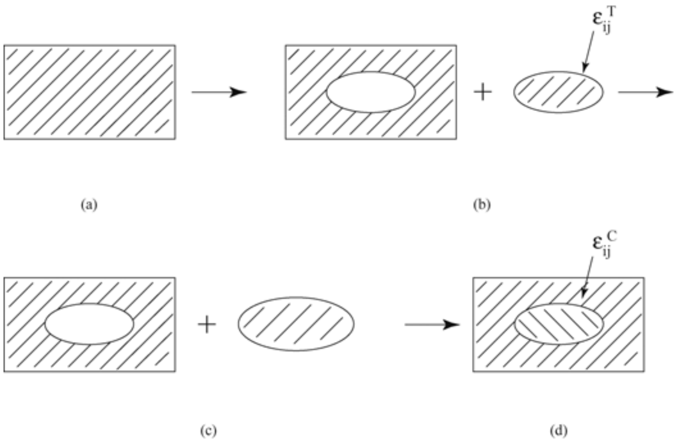

Fig. 11.1. Schematic illustration of homogenization according to the Eshelby method

the inclusion to return it to its original shape, then imagine returning it back to the matrix material (see Fig. 11.1c).
3. Imagine welding the inclusion and matrix material together then removing the surface tractions. The matrix and inclusion will then reach an equilibrium state when the inclusion has a *constraining strain* ε_{ij}^{C} relative to the initial shape before it was removed (see Fig. 11.1d).
4. The stress in the inclusion σ_{ij}^{I} can now be calculated as follows assuming the strain is uniform within the inclusion:

$$\sigma_{ij}^{I} = C_{ijkl}^{M} \left(\varepsilon_{kl}^{C} - \varepsilon_{kl}^{T} \right) \tag{11.1}$$

where C_{ijkl}^{M} are the components of the elasticity tensor of the matrix\material.
5. Eshelby has shown that the constraining strain ε_{ij}^{C} can be calculated in terms of the transformation strain ε_{ij}^{T} using the following equations:

$$\varepsilon_{ij}^{C} = S_{ijkl} \varepsilon_{kl}^{T} \tag{11.2}$$

where S_{ijkl} are the components of the Eshelby tensor **S**. The Eshelby tensor **S** is a fourth-rank tensor determined using Poisson's ratio of the inclusion material and the inclusion's aspect ration.
6. Finally, the stress in the inclusion is determined by substituting (11.2) into (11.1) and simplifying to obtain:

$$\sigma_{ij}^{I} = C_{ijkl}^{M} \left(S_{klmn} - I_{klmn} \right) \varepsilon_{mn}^{T} \tag{11.3}$$

where I_{klmn} are the components of the fourth-rank identity tensor given by:

$$I_{klmn} = \frac{1}{2}\left(\delta_{km}\delta_{\ln} + \delta_{kn}\delta_{lm}\right) \tag{11.4}$$

and δ_{ij} are the components of the Kronecker delta tensor.

Using matrices, (11.3) is re-written as follows:

$$\left\{\sigma^I\right\} = \left[C^M\right]\left([S] - [I]\right)\left\{\varepsilon^T\right\} \tag{11.5}$$

where the braces are used to indicate a vector while the brackets are used to indicate a matrix.

Next, expressions of the Eshelby tensor **S** are presented for the case of long infinite cylindrical fibers. In this case, the values of the Eshelby tensor depend on Poisson's ratio ν of the fibers and are determined as follows:

$$S_{1111} = S_{2222} = \frac{5-\nu}{8(1-\nu)} \tag{11.6a}$$

$$S_{3333} = 0 \tag{11.6b}$$

$$S_{1122} = S_{2211} = \frac{-1+4\nu}{8(1-\nu)} \tag{11.6c}$$

$$S_{1133} = S_{2233} = \frac{\nu}{2(1-\nu)} \tag{11.6d}$$

$$S_{3311} = S_{3322} = 0 \tag{11.6e}$$

$$S_{1212} = S_{1221} = S_{2112} = S_{2121} = \frac{3-4\nu}{8(1-\nu)} \tag{11.6f}$$

$$S_{1313} = S_{1331} = S_{3113} = S_{3131} = \frac{1}{4} \tag{11.6g}$$

$$S_{3232} = S_{3223} = S_{2332} = S_{2323} = \frac{1}{4} \tag{11.6h}$$

$$S_{ijkl} = 0\,, \qquad \text{otherwise} \tag{11.6i}$$

In addition to Eshelby's method of determining the stresses and strains in the fibers and matrix, there are other methods based on Hill's stress and strain concentration factors.

Problems

Problem 11.1

Derive the equations of the Eshelby method for the case of a misfit strain due to a differential thermal contraction assuming that the matrix and inclusion have different thermal expansion coefficients.

Problem 11.2

Derive the equations of the method for the case of internal stresses in externally loaded composites. Assume the existence of an external load that is responsible for the transfer of load to the inclusion.

Problem 11.3

The formulation in this chapter has been based on what are called dilute composite systems, i.e. a single inclusion is embedded within an infinite matrix. In this case, the inclusion volume fraction is less than a few percent. Consider non-dilute systems where the inclusion volume fraction is much higher with many inclusions. What modifications to the equations of the Eshelby method are needed to formulate the theory for non-dilute systems.

12

Introduction to Damage Mechanics of Composite Materials

12.1 Basic Equations

The objective of this final chapter is to introduce the reader to the subject of damage mechanics of composite materials. For further details, the reader is referred to the comprehensive book on this subject written by the authors: *Advances in Damage Mechanics: Metals and Metal Matrix Composites.* This chapter does not contain any MATLAB functions or examples.

In this chapter, only elastic composites are considered. The fibers are assumed to be continuous and perfectly aligned. In addition, a perfect bond is assumed to exist at the matrix-fiber interface. A consistent mathematical formulation is presented in the next sections to derive the equations of damage mechanics for these composite materials using two different approaches: one overall and one local. The elastic stiffness matrix is derived using both these two approaches and is shown to be identical in both cases.

For simplicity, the composite system is assumed to consist of a matrix reinforced with continuous fibers. Both the matrix and fibers are linearly elastic with different material constants. Let $\bar{C}$ denote the configuration of the undamaged composite system and let $\bar{C}^m$ and $\bar{C}^f$ denote the configurations of the undamaged matrix and fibers, respectively. Since the composite system assumes a perfect bond at the matrix-fiber interface, it is clear that $\bar{C}^m \cap \bar{C}^f = \phi$ and $\bar{C}^m \cup \bar{C}^f = \bar{C}$. In the overall approach, the problem reduces to transforming the undamaged configuration $\bar{C}$ into the damaged configuration C. In contrast, two intermediate configurations C^m and C^f are considered in the local approach for the matrix and fibers, respectively. In the latter approach, the problem is reduced to transforming each of the undamaged configurations $\bar{C}^m$ and $\bar{C}^f$ into the damaged configurations C^m and C^f, respectively.

In case of elastic fiber-reinforced composites, the following linear relation is used for each constituent in its respective undamaged configuration:

$$\bar{\sigma}^k = \bar{E}^k : \bar{\varepsilon}^k, \qquad k = m, f \tag{12.1}$$

where $\bar{\sigma}^k$ is the effective constituent stress tensor, $\bar{\varepsilon}^k$ is the effective strain tensor, and $\bar{E}^k$ is the effective constituent elasticity tensor. The operation: denotes the tensor contraction operation over two indices. For the case of an isotropic constituent, $\bar{E}^k$ is given by the following formula:

$$\bar{E}^k = \bar{\lambda}^k I_2 \otimes I_2 + 2\bar{u}^k I_4 \tag{12.2}$$

where $\bar{\lambda}^k$ and $\bar{\mu}^k$ are the effective constituent Lame's constants, I_2 is the second-rank identity tensor, and I_4 is the fourth-rank identity tensor. The operation $\otimes$ is the tensor cross product between two second-rank tensors to produce a fourth-rank tensor.

Within the framework of the micromechanical analysis of composite materials, the effective constituent stress tensor $\bar{\sigma}^k$ is related to the effective composite stress tensor $\bar{\sigma}$ by the following relation:

$$\bar{\sigma}^k = \bar{B}^k : \bar{\sigma} \tag{12.3}$$

The fourth-rank tensor $\bar{B}^k$ is the constituent *stress concentration tensor*. It can be determined using several available homogenization models such as the Voigt and Mori-Tanaka models. The effective constituent strain tensor $\bar{\varepsilon}^k$ is determined in a similar way by the following relation:

$$\bar{\varepsilon}^k = \bar{A}^k : \bar{\varepsilon} \tag{12.4}$$

where $\bar{\varepsilon}$ is the effective composite strain tensor and $\bar{A}^k$ is the fourth-rank constituent strain concentration tensor.

Next, the overall and local approaches to damage in elastic composites are examined in the following two sections.

12.2 Overall Approach

In this approach, damage is incorporated in the composite system as a whole through one damage tensor called the *overall damage tensor*. The two steps needed in this approach are shown schematically in Fig. 12.1 for a two-phase composite system consisting of a matrix and fibers. In the first step, the elastic equations are formulated in an undamaged composite system. This is performed here using the law of mixtures as follows:

$$\bar{\sigma} = \bar{c}^m \bar{\sigma}^m + \bar{c}^f \bar{\sigma}^f \tag{12.5}$$

where $\bar{c}^m$ and $\bar{c}^f$ are the effective matrix and fiber volume fractions, respectively.

In the effective composite configuration $\bar{C}$, the following linear elastic relation holds:

$$\bar{\sigma} = \bar{E} : \bar{\varepsilon} \tag{12.6}$$

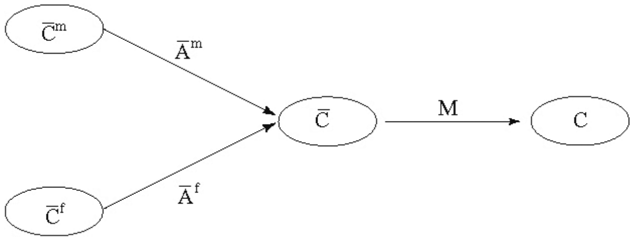

Fig. 12.1. Schematic diagram illustrating the overall approach for composite materials

where $\bar{E}$ is the fourth-rank constant elasticity tensor. Substituting (12.1), (12.4), and (12.6) into (12.5) and simplifying, one obtains the following expression for $\bar{E}$:

$$\bar{E} = \bar{c}^m \bar{E}^m : \bar{A}^m + \bar{c}^f \bar{E}^f : \bar{A}^f \tag{12.7}$$

In the second step of the formulation, damage is induced through the fourth-rank damage effect tensor M as follows:

$$\bar{\sigma} = M : \sigma \tag{12.8}$$

where σ is the composite stress tensor. Equation (12.8) represents the damage transformation equation for the stress tensor. In order to derive a similar relation for the strain tensor, one needs to use the *hypothesis of elastic energy equivalence*. In this hypothesis, the elastic energy of the damage system is equal to the elastic energy of the effective system. Applying this hypothesis to the composite system by equating the two elastic energies, one obtains:

$$\frac{1}{2}\varepsilon : \sigma = \frac{1}{2}\bar{\varepsilon} : \bar{\sigma} \tag{12.9}$$

where ε is the composite strain tensor. Substituting (12.8) into (12.9) and simplifying, one obtains the damage transformation equation for the strain tensor as follows:

$$\bar{\varepsilon} = M^{-T} : \varepsilon \tag{12.10}$$

where the superscript $-T$ denotes the inverse transpose of the tensor.

In order to derive the final relation in the damaged composite system, one substitutes (12.8) and (12.10) into (12.6) to obtain:

$$\sigma = E : \varepsilon \tag{12.11}$$

where the fourth-rank elasticity tensor E is given by:

$$E = M^{-1} : \bar{E} : M^{-T} \tag{12.12a}$$

Substituting for $\bar{E}$ from (12.7) into (12.12a), one obtains:

$$E = M^{-1} : \left(\bar{c}^m \bar{E}^m : \bar{A}^m + \bar{c}^f \bar{E}^f : \bar{A}^f\right) : M^{-T} \tag{12.12b}$$

The above equation represents the elasticity tensor in the damaged composite system according to the overall approach.

12.3 Local Approach

In this approach, damage is introduced in the first step of the formulation using two independent damage tensors for the matrix and fibers. However, more damage tensors may be introduced to account for other types of damage such as debonding and delamination. The two steps involved in this approach are shown schematically in Fig. 12.2. One first introduces the fourth-rank *matrix and fiber damage effect tensors* M^m and M^f, respectively, as follows:

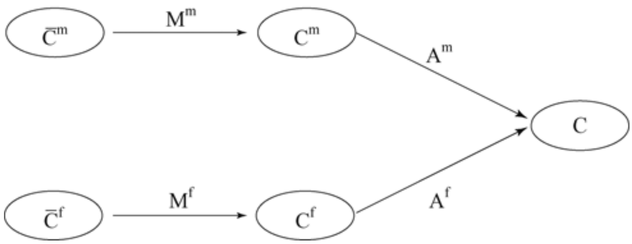

Fig. 12.2. Schematic diagram illustrating the local approach for composite materials

$$\bar{\sigma}^k = M^k : \sigma^k \ , \qquad k = m, f \tag{12.13}$$

The above equation can be interpreted in a similar way to (12.8) except that it applies at the constituent level. It also represents the damage transformation equation for each constituent stress tensor. In order to derive a similar transformation equation for the constituent strain tensor, one applies the hypothesis of elastic energy equivalence to each constituent separately as follows:

$$\frac{1}{2}\varepsilon^k : \sigma^k = \frac{1}{2}\bar{\varepsilon}^k : \bar{\sigma}^k \ , \qquad k = m, f \tag{12.14}$$

In using (12.14), one assumes that there are no micromechanical or constituent elastic interactions between the matrix and fibers. This assumption is not valid in general. From micromechanical considerations, there should be interactions between the elastic energies in the matrix and fibers. However, such interactions are beyond the scope of this book as the resulting equations will be complicated and the sought relations may consequently be unattainable. It should be clear to the reader that (12.14) is the single most important assumption that is needed to derive the relations of the local approach. It will also be needed later when we show the equivalence of the overall and local approaches. Therefore, the subsequent relations are very special cases when (12.14) is valid.

Substituting (12.13) into (12.14) and simplifying, one obtains the required transformations for the constituent strain tensor as follows:

$$\bar{\varepsilon}^k = M^{k^{-T}} : \varepsilon^k \ , \qquad k = m, f \tag{12.15}$$

The above equation implies a decoupling between the elastic energy in the matrix and fibers. Other methods may be used that include some form of coupling but they will lead to complicated transformation equations that are beyond the scope of this book.

Substituting (12.13) and (12.15) into (12.1) and simplifying, one obtains:

$$\sigma^k = E^k : \varepsilon^k \ , \qquad k = m, f \tag{12.16}$$

where the constituent elasticity tensor E^k is given by:

$$E^k = M^{k^{-1}} : \bar{E}^k : M^{k^{-T}} \ , \qquad k = m, f \tag{12.17}$$

Equation (12.16) represents the elasticity relation for the damaged constituents. The second step of the formulation involves transforming (12.17) into the whole composite system using the law of mixtures as follows:

$$\sigma = c^m \sigma^m + c^f \sigma^f \tag{12.18}$$

where c^m and c^f are the matrix and fiber volume fractions, respectively, in the damaged composite system. Before proceeding with (12.18), one needs to derive a strain constituent equation similar to (12.4). Substituting (12.10) and (12.15) into (12.4) and simplifying, one obtains:

$$\varepsilon^k = A^k : \varepsilon \ , \qquad k = m, f \tag{12.19}$$

where the constituent strain concentration tensor A^k in the damaged state is given by:

$$A^k = M^{k^T} : \bar{A}^k : M^{-T} \ , \qquad k = m, f \tag{12.20}$$

The above equation represents the damage transformation equation for the strain concentration tensor.

Finally, one substitutes (12.11), (12.16), and (12.19) into (12.18) and simplifies to obtain:

$$E = c^m E^m : A^m + c^f E^f : A^f \tag{12.21}$$

Equation (12.21) represents the elasticity tensor in the damaged composite system according to the local approach.

12.4 Final Remarks

In this final section, it is shown that both the overall and local approaches are equivalent elastic composites which are considered here. This proof is performed by showing that both the elasticity tensors given in (12.12b) and (12.21) are exactly the same. In fact, it is shown that (12.21) reduces to (12.12b) after making the appropriate substitution.

First, one needs to find a damage transformation equation for the volume fractions. This is performed by substituting (12.8) and (12.13) into (12.5), simplifying and comparing the result with (12.18). One therefore obtains:

$$c^k I_4 = \bar{c}^k M^{-1} : M^k \ , \qquad k = m, f \tag{12.22}$$

where I_4 is the fourth-rank identity tensor. Substituting (12.17) and (12.20) into (12.21) and simplifying, one obtains:

$$E = \left(c^m M^{m^{-1}} : \bar{E}^m : \bar{A}^m + c^f M^{f^{-1}} : \bar{E}^f : \bar{A}^f \right) : M^{-T} \tag{12.23}$$

Finally, one substitutes (12.22) into (12.23) and simplifies to obtain:

$$E = M^{-1} : \left(\bar{c}^m \bar{E}^m : \bar{A}^m + \bar{c}^f \bar{E}^f : \bar{A}^f \right) : M^{-T} \tag{12.24}$$

It is clear that the above equation is the same as (12.12b). Therefore, both the overall and local approaches yield the same elasticity tensor in the damaged composite system.

Equation (12.24) can be generalized to an elastic composite system with n constituents as follows:

$$E = M^{-1} : \left(\sum_{k=1}^{n} \bar{c}^k \bar{E}^k : \bar{A}^k \right) : M^{-T} \tag{12.25}$$

The two formulations of the overall and local approaches can be used to obtain the above equation for a composite system with n constituents. The derivation of (12.25) is similar to the derivation of (12.24) – therefore it is not presented here and is left to the problems.

In the remaining part of this section, some additional relations are presented to relate the overall damage effect tensor with the constituent damage effect tensors. Substituting (12.3) into (12.5) and simplifying, one obtains the constraint equation for the stress concentration tensors. The constraint equation is generalized as follows:

$$\sum_{k=1}^{n} \bar{c}^k \bar{B}^k = I_4 \tag{12.26}$$

where I_4 is the fourth-rank identity tensor. To find a relation between the stress concentration tensors in the effective and damaged states, one substitutes (12.8) and (12.13) into (12.3) and simplifies to obtain:

$$\sigma^k = B^k : \sigma \ , \qquad k = 1, 2, 3, \ldots, n \tag{12.27}$$

where B^k is the fourth-rank stress concentration tensor in the damaged configuration and is given by:

$$B^k = M^{k^{-1}} : \bar{B}^k : M \ , \qquad k = 1, 2, 3, \ldots, n \tag{12.28}$$

Substituting (12.27) into (12.18) and simplifying, the resulting constraint is generalized as follows:

$$\sum_{k=1}^{n} c^k B^k = I_4 \tag{12.29}$$

Finally, substituting (12.28) into (12.29) and simplifying, one obtains:

$$M = \left(\sum_{k=1}^{n} c^k M^{k^{-1}} : \bar{B}^k \right)^{-1} \tag{12.30}$$

Equation (12.30) represents the required relation between the overall and local (constituent) damage effect tensors.

Problems

Problem 12.1

Consider a composite system that consists of n constituents. In this case, the overall approach is schematically illustrated in Fig. 12.3. In this case, derive (12.25) in detail.

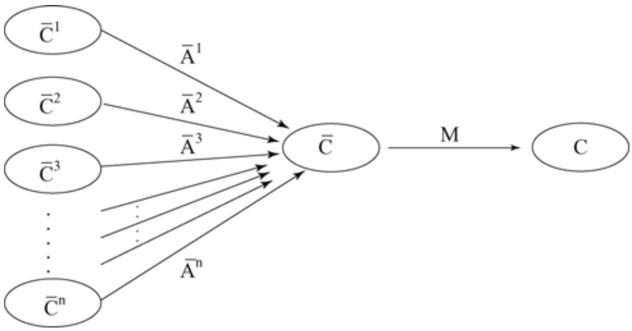

Fig. 12.3. Schematic diagram illustrating the overall approach for composite materials for Problem 12.1

Problem 12.2

Consider a composite system that consists of n constituents. In this case, the local approach is schematically illustrated in Fig. 12.4. In this case, derive (12.25) in detail.

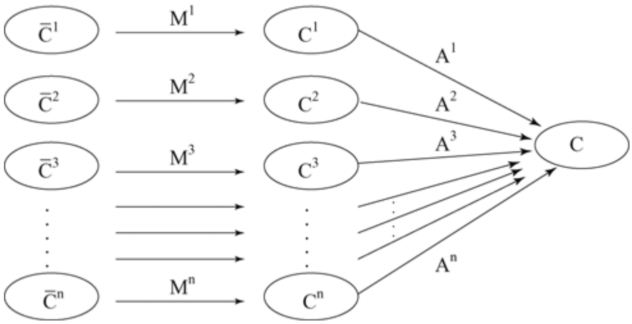

Fig. 12.4. Schematic diagram illustrating the local approach for composite materials for Problem 12.2

Problem 12.3

Derive (12.20) in detail.

Problem 12.4

Derive (12.28) in detail.

Problem 12.5

The stress and strain concentration tensors are usually determined using one of the following four models:

1. The Voigt model.
2. The Reuss model.
3. The Mori-Tanaka model.
4. The Eshelby Tensor.

Make a literature search on the above four models and describe each model briefly writing its basic equations.

Solutions to Problems

Problem 2.1

In this case, $[S]$ is symmetric given as follows:

$$[S] = \begin{bmatrix} S_{11} & S_{12} & S_{13} & 0 & 0 & 0 \\ S_{12} & S_{22} & S_{23} & 0 & 0 & 0 \\ S_{13} & S_{23} & S_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & S_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & S_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & S_{66} \end{bmatrix}$$

$$\begin{aligned} |S| &= [S_{11}(S_{22}S_{33} - S_{23}S_{23}) - S_{12}(S_{12}S_{33} - S_{13}S_{23}) \\ &\quad +S_{13}(S_{12}S_{23} - S_{13}S_{22})]\, S_{44}S_{55}S_{66} \\ &= (S_{11}S_{22}S_{33} - S_{11}S_{23}S_{23} - S_{33}S_{12}S_{12} \\ &\quad -S_{22}S_{13}S_{13} + 2S_{12}S_{23}S_{13})\, S_{44}S_{55}S_{66} \end{aligned}$$

Next, use the following formula to calculate the inverse of $[S]$:

$$[C] = [S]^{-1} = \frac{adj[S]}{|S|}$$

Only C_{11} will be calculated in detail as follows:

$$C_{11} = \frac{(adj[S])_{11}}{|S|} = \frac{(S_{22}S_{33} - S_{23}S_{23})\, S_{44}S_{55}S_{66}}{|S|} = \frac{1}{S}(S_{22}S_{33} - S_{23}S_{23})$$

where S is given in the book in (2.5). The same procedure can be followed to derive the other elements of $[C]$ given in (2.5).

Problem 2.2

The reciprocity relations of (2.6) are valid for linear elastic analysis. They can be derived by applying the Maxwell-Betti Reciprocal Theorem. For more details, see [1].

Problem 2.3

$$[S] = \begin{bmatrix} 1/E_1 & -\nu_{12}/E_1 & -\nu_{13}/E_1 & 0 & 0 & 0 \\ -\nu_{12}/E_1 & 1/E_2 & -\nu_{23}/E_2 & 0 & 0 & 0 \\ -\nu_{13}/E_1 & -\nu_{23}/E_2 & 1/E_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \dfrac{2(1+\nu_{23})}{E_2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/G_{12} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/G_{12} \end{bmatrix}$$

Problem 2.4

$$\begin{aligned} S &= S_{11}S_{22}S_{33} - S_{11}S_{23}S_{23} - S_{22}S_{13}S_{13} - S_{33}S_{12}S_{12} + 2S_{12}S_{23}S_{13} \\ &= \frac{1}{E_1}\frac{1}{E_2}\frac{1}{E_2} - \frac{1}{E_1}\left(\frac{-\nu_{23}}{E_2}\right)\left(\frac{-\nu_{23}}{E_2}\right) - \frac{1}{E_2}\left(\frac{-\nu_{12}}{E_1}\right)\left(\frac{-\nu_{21}}{E_2}\right) \\ &\quad - \frac{1}{E_2}\left(\frac{-\nu_{12}}{E_1}\right)\left(\frac{-\nu_{21}}{E_2}\right) + 2\left(\frac{-\nu_{12}}{E_1}\right)\left(\frac{-\nu_{23}}{E_2}\right)\left(\frac{-\nu_{21}}{E_2}\right) \\ &= \frac{1 - \nu_{23}^2 - 2\nu_{12}\nu_{21} - 2\nu_{12}\nu_{23}\nu_{21}}{E_1 E_2^2} \\ &= \frac{1-\nu'}{E_1 E_2^2} \end{aligned}$$

where ν' is given by:

$$\nu' = \nu_{23}^2 + 2\nu_{12}\nu_{21} + 2\nu_{12}\nu_{23}\nu_{21}$$

Next, C_{11} is calculated in detail as follows:

$$\begin{aligned} C_{11} &= \frac{1}{S}(S_{22}S_{33} - S_{23}S_{23}) \\ &= \frac{E_1 E_2^2}{1-\nu'}\left[\frac{1}{E_2}\frac{1}{E_2} - \left(\frac{-\nu_{23}}{E_2}\right)\left(\frac{-\nu_{23}}{E_2}\right)\right] \\ &= \frac{\left(1-\nu_{23}^2\right)E_1}{1-\nu'} \end{aligned}$$

Similarly, the other elements of $[C]$ are obtained as follows:

$$
\begin{aligned}
C_{12} &= \frac{(1+\nu_{23})\nu_{12}E_2}{1-\nu'} \\
C_{13} &= \frac{(1+\nu_{23})\nu_{12}E_2}{1-\nu'} = C_{12} \\
C_{22} &= \frac{(1-\nu_{12}\nu_{21})E_2}{1-\nu'} \\
C_{23} &= \frac{(\nu_{23}+\nu_{12}\nu_{21})E_2}{1-\nu'} \\
C_{33} &= \frac{(1-\nu_{12}\nu_{21})E_2}{1-\nu'} = C_{22} \\
C_{44} &= \frac{E_2}{2(1+\nu_{23})} \\
C_{55} &= G_{12} \\
C_{66} &= G_{12} = C_{55}
\end{aligned}
$$

Problem 2.5

$$
[S] = \begin{bmatrix}
\frac{1}{E} & \frac{-\nu}{E} & \frac{-\nu}{E} & 0 & 0 & 0 \\
\frac{-\nu}{E} & \frac{1}{E} & \frac{-\nu}{E} & 0 & 0 & 0 \\
\frac{-\nu}{E} & \frac{-\nu}{E} & \frac{1}{E} & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{2(1+\nu)}{E} & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{2(1+\nu)}{E} & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{2(1+\nu)}{E}
\end{bmatrix}
$$

$$
[S] = \frac{1}{E}\begin{bmatrix}
1 & -\nu & -\nu & 0 & 0 & 0 \\
-\nu & 1 & -\nu & 0 & 0 & 0 \\
-\nu & -\nu & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 2(1+\nu) & 0 & 0 \\
0 & 0 & 0 & 0 & 2(1+\nu) & 0 \\
0 & 0 & 0 & 0 & 0 & 2(1+\nu)
\end{bmatrix}
$$

Problem 2.6

$$[C] = \frac{E}{(1+\nu)(1+2\nu)} \begin{bmatrix} 1 & 1-\nu & 1-\nu & 0 & 0 & 0 \\ 1-\nu & 1 & 1-\nu & 0 & 0 & 0 \\ 1-\nu & 1-\nu & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \dfrac{1+2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \dfrac{1+2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \dfrac{1+2\nu}{2} \end{bmatrix}$$

Problem 2.7

```
>> sigma3 = 150/(40*40)

sigma3 =

   0.0938

>> sigma = [0 ; 0 ; sigma3 ; 0 ; 0 ; 0]

sigma =

        0
        0
   0.0938
        0
        0
        0

>> [S] =
OrthotropicCompliance(50.0,15.2,15.2,0.254,0.428,0.254,4.70,
   3.28,4.70)

S =
    0.0200   -0.0051   -0.0051         0         0         0
   -0.0051    0.0658   -0.0282         0         0         0
   -0.0051   -0.0282    0.0658         0         0         0
         0         0         0    0.3049         0         0
         0         0         0         0    0.2128         0
         0         0         0         0         0    0.2128

>> epsilon = S*sigma
```

```
epsilon =

   -0.0005
   -0.0026
    0.0062
         0
         0
         0

>> format short e
>> epsilon

epsilon =

   -4.7625e-004
   -2.6398e-003
    6.1678e-003
              0
              0
              0

>> d1 = epsilon(1)*40

d1 =

   -1.9050e-002

>> d2 = epsilon(2)*40

d2 =

   -1.0559e-001

>> d3 = epsilon(3)*40

d3 =

   2.4671e-001
```

Problem 2.8

```
>> sigma3 = 150/(40*40)

sigma3 =

    0.0938
```

```
>> sigma = [0 ; 0 ; sigma3 ; 0 ; 0 ; 0]

sigma =

         0
         0
    0.0938
         0
         0
         0

>> [S] = IsotropicCompliance(72.4,0.3)

S =

    0.0138   -0.0041   -0.0041         0         0         0
   -0.0041    0.0138   -0.0041         0         0         0
   -0.0041   -0.0041    0.0138         0         0         0
         0         0         0    0.0359         0         0
         0         0         0         0    0.0359         0
         0         0         0         0         0    0.0359

>> epsilon = S*sigma

epsilon =
   -0.0004
   -0.0004
    0.0013
         0
         0
         0

>> format short e
>> epsilon

epsilon =

   -3.8847e-004
   -3.8847e-004
    1.2949e-003
              0
              0
              0

>> d1 = epsilon(1)*40

d1 =

   -1.5539e-002
```

```
>> d2 = epsilon(2)*40

d2 =

   -1.5539e-002

>> d3 = epsilon(3)*40

d3 =

    5.1796e-002
```

Problem 2.9

```
>> sigma2 = 100/(60*60)

sigma2 =

    0.0278

>> sigma = [0 ; sigma2 ; 0 ; 0 ; 0 ; 0]

sigma =

         0
    0.0278
         0
         0
         0
         0

>> [S] =

OrthotropicCompliance(155.0,12.10,12.10,0.248,0.458,0.248,
   4.40,3.20,4.40)

S =

     0.0065   -0.0016   -0.0016         0         0         0
    -0.0016    0.0826   -0.0379         0         0         0
    -0.0016   -0.0379    0.0826         0         0         0
          0         0         0    0.3125         0         0
          0         0         0         0    0.2273         0
          0         0         0         0         0    0.2273
```

```
>> EpsilonMechanical = S*sigma

EpsilonMechanical =

   -0.0000
    0.0023
   -0.0011
         0
         0
         0

>> format short e
>> EpsilonMechanical

EpsilonMechanical =

   -4.4444e-005
    2.2957e-003
   -1.0514e-003
              0
              0
              0

>> EpsilonThermal(1) = -0.01800e-6*30

EpsilonThermal =

   -5.4000e-007

>> EpsilonThermal(2) = 24.3e-6*30

EpsilonThermal =

   -5.4000e-007  7.2900e-004

>> EpsilonThermal(3) = 24.3e-6*30

EpsilonThermal =

   -5.4000e-007  7.2900e-004  7.2900e-004

>> EpsilonThermal(4) = 0

EpsilonThermal =

   -5.4000e-007  7.2900e-004  7.2900e-004            0

>> EpsilonThermal(5) = 0
```

```
EpsilonThermal =

   -5.4000e-007  7.2900e-004  7.2900e-004            0            0

>> EpsilonThermal(6) = 0

EpsilonThermal =

   -5.4000e-007  7.2900e-004  7.2900e-004            0            0            0

>> EpsilonThermal = EpsilonThermal'

EpsilonThermal =

   -5.4000e-007
    7.2900e-004
    7.2900e-004
              0
              0
              0

>> Epsilon = EpsilonMechanical + EpsilonThermal

Epsilon =

   -4.4984e-005
    3.0247e-003
   -3.2242e-004
              0
              0
              0

>> d1 = Epsilon(1)*60

d1 =

  -2.6991e-003

>> d2 = Epsilon(2)*60

d2 =

   1.8148e-001

>> d3 = Epsilon(3)*60
```

```
d3 =

  -1.9345e-002

>>
```

Problem 2.10

$$\begin{Bmatrix} \varepsilon_1 - \alpha_1\Delta T - \beta_1\Delta T \\ \varepsilon_2 - \alpha_2\Delta T - \beta_2\Delta T \\ \varepsilon_3 - \alpha_3\Delta T - \beta_3\Delta T \\ \gamma_{23} \\ \gamma_{13} \\ \gamma_{12} \end{Bmatrix} = \begin{bmatrix} S_{11} & S_{12} & S_{13} & 0 & 0 & 0 \\ S_{12} & S_{22} & S_{23} & 0 & 0 & 0 \\ S_{13} & S_{23} & S_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & S_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & S_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & S_{66} \end{bmatrix} \begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \tau_{23} \\ \tau_{13} \\ \tau_{12} \end{Bmatrix}$$

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \tau_{23} \\ \tau_{13} \\ \tau_{12} \end{Bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & 0 & 0 & 0 \\ C_{12} & C_{22} & C_{23} & 0 & 0 & 0 \\ C_{13} & C_{23} & C_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_1 - \alpha_1\Delta T - \beta_1\Delta T \\ \varepsilon_2 - \alpha_2\Delta T - \beta_2\Delta T \\ \varepsilon_3 - \alpha_3\Delta T - \beta_3\Delta T \\ \gamma_{23} \\ \gamma_{13} \\ \gamma_{12} \end{Bmatrix}$$

Problem 3.1

Let A be the total cross-sectional area of the unit cell and let A^f and A^m be the cross-sectional areas of the fiber and matrix, respectively. Then, we have the following relations based on the geometry of the problem:

$$A^f + A^m = A$$

Divide both sides of the above equation by A to obtain:

$$\frac{A^f}{A} + \frac{A^m}{A} = 1$$

Substituting $A^f/A = V^f$ and $A^m/A = V^m$, we obtain (3.1) as follows:

$$V^f + V^m = 1$$

Problem 3.2

Let W be the width of the cross-section in Fig. 3.3 (see book). Also, let W^f and W^m be the widths of the fiber and matrix, respectively.

$$\begin{aligned}
\nu_{12}^f &= -\frac{\Delta W^f/W^f}{\Delta L/L} \\
\nu^m &= -\frac{\Delta W^m/W^m}{\Delta L/L} \\
\Delta W^f &= -\nu_{12}^f W^f \frac{\Delta L}{L} \\
\Delta W^m &= -\nu^m W^m \frac{\Delta L}{L} \\
\Delta W &= \Delta W^f + \Delta W^m \\
&= -\left(\nu_{12}^f W^f + \nu^m W^m\right) \frac{\Delta L}{L} \\
\frac{\Delta W}{W} &= -\left(\nu_{12}^f \frac{W^f}{W} + \nu^m \frac{W^m}{W}\right) \frac{\Delta L}{L} \\
-\frac{\Delta W/W}{\Delta L/L} &= \nu_{12}^f V^f + \nu^m V^m
\end{aligned}$$

where $W^f/W = V^f$ and $W^m/W = V^m$. Then, we obtain:

$$\nu_{12} = \nu_{12}^f V^f + \nu^m V^m$$

Problem 3.3

Let W be the width of the cross-section in Fig. 3.3 (see book). Also, let W^f and W^m be the widths of the fiber and matrix, respectively. Also, from equilibrium, we have $\sigma_2^f = \sigma_2^m = \sigma_2$.

$$\begin{aligned}
\sigma_2^f &= \sigma_2 = E_2^f \varepsilon_2^f = E_2^f \frac{\Delta W^f}{W^f} \\
\sigma_2^m &= \sigma_2 = E^m \varepsilon_2^m = E^m \frac{\Delta W^m}{W^m} \\
\Delta W^f &= \frac{W^f}{E_2^f} \sigma_2 \\
\Delta W^m &= \frac{W^m}{E^m} \sigma_2 \\
\varepsilon_2 &= \frac{\Delta W}{W} = \frac{\Delta W^f + \Delta W^m}{W}
\end{aligned}$$

$$= \frac{\left(\frac{W^f}{E_2^f} + \frac{W^m}{E^m}\right)\sigma_2}{W}$$

$$\varepsilon_2 = \left(\frac{W^f/W}{E_2^f} + \frac{W^m/W}{E^m}\right)\sigma_2$$

$$\varepsilon_2 = \frac{1}{E_2}\sigma_2$$

$$\frac{1}{E_2} = \frac{V^f}{E_2^f} + \frac{V^m}{E^m}$$

where $W^f/W = V^f$ and $W^m/W = V^m$.

Problem 3.4

The following is a listing of the modified MATLAB function $E2$ called *E2Modified*. Note that this modified function is available with the M-files for the book on the CD-ROM that accompanies the book.

```
function y = E2Modified(Vf,E2f,Em,Eta,NU12f,NU21f,NUm,E1f,p)
%E2Modified  This function returns Young's modulus in the
%            transverse direction. Its input are nine values:
%            Vf    - fiber volume fraction
%            E2f   - transverse Young's modulus of the fiber
%            Em    - Young's modulus of the matrix
%            Eta   - stress-partitioning factor
%            NU12f - Poisson's ratio NU12 of the fiber
%            NU21f - Poisson's ratio NU21 of the fiber
%            NUm   - Poisson's ratio of the matrix
%            E1f   - longitudinal Young's modulus of the fiber
%            p     - parameter used to determine which equation to use:
%                  p = 1 - use equation (3.4)
%                  p = 2 - use equation (3.9)
%                  p = 3 - use equation (3.10)
%                  p = 4 - use the modified formula using (3.23)
%            Use the value zero for any argument not needed
%            in the calculations.
Vm = 1 - Vf;
if p == 1
     y = 1/(Vf/E2f + Vm/Em);
elseif p == 2
     y = 1/((Vf/E2f + Eta*Vm/Em)/(Vf + Eta*Vm));
elseif p == 3
     deno = E1f*Vf + Em*Vm;
     etaf = (E1f*Vf + ((1-NU12f*NU21f)*Em + NUm*NU21f
            *E1f)*Vm)/deno;
```

```
    etam = (((1-NUm*NUm)*E1f - (1-NUm*NU12f)*Em)*Vf
           + Em*Vm)/deno;
    y = 1/(etaf*Vf/E2f + etam*Vm/Em);
elseif p == 4
    EmPrime = Em/(1 - NUm*NUm);
    y = 1/(Vf/E2f + Vm/EmPrime);
end
```

Problem 3.5

The transverse modulus E_2 is calculated in GPa using the three different formulas with the MATLAB function $E2$ as follows. Note that the three values obtained are comparable and very close to each other.

```
>> E2(0.65, 14.8, 3.45, 0, 0, 0, 0, 0, 1)

ans =

   6.8791

>> E2(0.65, 14.8, 3.45, 0.5, 0, 0, 0, 0, 2)

ans =

   8.7169

>> E2(0.65, 14.8, 3.45, 0, 0.3, 0.3, 0.36, 85.6, 3)

ans =

   7.6135
```

Problem 3.6

```
>> y(1) = E2(0, 14.8, 3.45, 0, 0, 0, 0, 0, 1)

y =

  3.4500

>> y(2) = E2(0.1, 14.8, 3.45, 0, 0, 0, 0, 0, 1)

y =

  3.4500   3.7366
```

```
>> y(3) = E2(0.2, 14.8, 3.45, 0, 0, 0, 0, 0, 1)

y =

  3.4500  3.7366  4.0750

>> y(4) = E2(0.3, 14.8, 3.45, 0, 0, 0, 0, 0, 1)

y =

  3.4500  3.7366  4.0750  4.4809

>> y(5) = E2(0.4, 14.8, 3.45, 0, 0, 0, 0, 0, 1)

y =

  3.4500  3.7366  4.0750  4.4809  4.9766

>> y(6) = E2(0.5, 14.8, 3.45, 0, 0, 0, 0, 0, 1)

y =

  3.4500  3.7366  4.0750  4.4809  4.9766  5.5956

>> y(7) = E2(0.6, 14.8, 3.45, 0, 0, 0, 0, 0, 1)

y =

  3.4500  3.7366  4.0750  4.4809  4.9766  5.5956  6.3905

>> y(8) = E2(0.7, 14.8, 3.45, 0, 0, 0, 0, 0, 1)

y =

  3.4500  3.7366  4.0750  4.4809  4.9766  5.5956  6.3905
  7.4486

>> y(9) = E2(0.8, 14.8, 3.45, 0, 0, 0, 0, 0, 1)

y =

  3.4500  3.7366  4.0750  4.4809  4.9766  5.5956  6.3905
  7.4486  8.9266
```

```
>> y(10) = E2(0.9, 14.8, 3.45, 0, 0, 0, 0, 0, 1)

y =

  3.4500  3.7366  4.0750  4.4809  4.9766  5.5956  6.3905
  7.4486  8.9266  11.1363

>> y(11) = E2(1, 14.8, 3.45, 0, 0, 0, 0, 0, 1)

y =

  3.4500  3.7366  4.0750  4.4809  4.9766  5.5956  6.3905
  7.4486  8.9266  11.1363  14.8000

>> z(1) = E2(0, 14.8, 3.45, 0.4, 0, 0, 0, 0, 2)

z =

  3.4500

>> z(2) = E2(0.1, 14.8, 3.45, 0.4, 0, 0, 0, 0, 2)

z =

  3.4500  4.1402

>> z(3) = E2(0.2, 14.8, 3.45, 0.4, 0, 0, 0, 0, 2)

z =

  3.4500  4.1402  4.8933

>> z(4) = E2(0.3, 14.8, 3.45, 0.4, 0, 0, 0, 0, 2)

z =

  3.4500  4.1402  4.8933  5.7182

>> z(5) = E2(0.4, 14.8, 3.45, 0.4, 0, 0, 0, 0, 2)

z =

  3.4500  4.1402  4.8933  5.7182  6.6258

>> z(6) = E2(0.5, 14.8, 3.45, 0.4, 0, 0, 0, 0, 2)

z =

  3.4500  4.1402  4.8933  5.7182  6.6258  7.6290
```

```
>> z(7) = E2(0.6, 14.8, 3.45, 0.4, 0, 0, 0, 0, 2)

z =

  3.4500  4.1402  4.8933  5.7182  6.6258  7.6290  8.7439


>> z(8) = E2(0.7, 14.8, 3.45, 0.4, 0, 0, 0, 0, 2)

z =

  3.4500  4.1402  4.8933  5.7182  6.6258  7.6290  8.7439
  9.9903

>> z(9) = E2(0.8, 14.8, 3.45, 0.4, 0, 0, 0, 0, 2)

z =

  3.4500  4.1402  4.8933  5.7182  6.6258  7.6290  8.7439
  9.9903  11.3927

>> z(10) = E2(0.9, 14.8, 3.45, 0.4, 0, 0, 0, 0, 2)

z =

  3.4500  4.1402  4.8933  5.7182  6.6258  7.6290  8.7439
  9.9903  11.3927  12.9825

>> z(11) = E2(1, 14.8, 3.45, 0.4, 0, 0, 0, 0, 2)

z =

  3.4500  4.1402  4.8933  5.7182  6.6258  7.6290  8.7439
  9.9903  11.3927  12.9825  14.8000
>> w(1) = E2(0, 14.8, 3.45, 0.5, 0, 0, 0, 0, 2)

w =

  3.4500

>> w(2) = E2(0.1, 14.8, 3.45, 0.5, 0, 0, 0, 0, 2)

w =

  3.4500  4.0090

>> w(3) = E2(0.2, 14.8, 3.45, 0.5, 0, 0, 0, 0, 2)
```

```
w =

  3.4500  4.0090  4.6348

>> w(4) = E2(0.3, 14.8, 3.45, 0.5, 0, 0, 0, 0, 2)


w =

  3.4500  4.0090  4.6348  5.3401

>> w(5) = E2(0.4, 14.8, 3.45, 0.5, 0, 0, 0, 0, 2)

w =

  3.4500  4.0090  4.6348  5.3401  6.1412

>> w(6) = E2(0.5, 14.8, 3.45, 0.5, 0, 0, 0, 0, 2)

w =

  3.4500  4.0090  4.6348  5.3401  6.1412  7.0590

>> w(7) = E2(0.6, 14.8, 3.45, 0.5, 0, 0, 0, 0, 2)

w =

  3.4500  4.0090  4.6348  5.3401  6.1412  7.0590  8.1209

>> w(8) = E2(0.7, 14.8, 3.45, 0.5, 0, 0, 0, 0, 2)

w =

  3.4500  4.0090  4.6348  5.3401  6.1412  7.0590  8.1209
  9.3638

>> w(9) = E2(0.8, 14.8, 3.45, 0.5, 0, 0, 0, 0, 2)

w =

  3.4500  4.0090  4.6348  5.3401  6.1412  7.0590  8.1209
  9.3638  10.8382

>> w(10) = E2(0.9, 14.8, 3.45, 0.5, 0, 0, 0, 0, 2)
```

```
w =

  3.4500  4.0090  4.6348  5.3401  6.1412  7.0590  8.1209
  9.3638  10.8382  12.6156

>> w(11) = E2(1, 14.8, 3.45, 0.5, 0, 0, 0, 0, 2)

w =

  3.4500  4.0090  4.6348  5.3401  6.1412  7.0590  8.1209
  9.3638  10.8382  12.6156  14.8000

>> u(1) = E2(0, 14.8, 3.45, 0.6, 0, 0, 0, 0, 2)

u =

  3.4500

>> u(2) = E2(0.1, 14.8, 3.45, 0.6, 0, 0, 0, 0, 2)

u =

  3.4500  3.9197

>> u(3) = E2(0.2, 14.8, 3.45, 0.6, 0, 0, 0, 0, 2)

u =

  3.4500  3.9197  4.4548

>> u(4) = E2(0.3, 14.8, 3.45, 0.6, 0, 0, 0, 0, 2)

u =

  3.4500  3.9197  4.4548  5.0701

>> u(5) = E2(0.4, 14.8, 3.45, 0.6, 0, 0, 0, 0, 2)

u =

  3.4500  3.9197  4.4548  5.0701  5.7850

>> u(6) = E2(0.5, 14.8, 3.45, 0.6, 0, 0, 0, 0, 2)

u =

  3.4500  3.9197  4.4548  5.0701  5.7850  6.6258
```

```
>> u(7) = E2(0.6, 14.8, 3.45, 0.6, 0, 0, 0, 0, 2)

u =

  3.4500  3.9197  4.4548  5.0701  5.7850  6.6258  7.6290

>> u(8) = E2(0.7, 14.8, 3.45, 0.6, 0, 0, 0, 0, 2)

u =

  3.4500  3.9197  4.4548  5.0701  5.7850  6.6258  7.6290
  8.8468

>> u(9) = E2(0.8, 14.8, 3.45, 0.6, 0, 0, 0, 0, 2)

u =

  3.4500  3.9197  4.4548  5.0701  5.7850  6.6258  7.6290
  8.8468  10.3561

>> u(10) = E2(0.9, 14.8, 3.45, 0.6, 0, 0, 0, 0, 2)

u =

  3.4500  3.9197  4.4548  5.0701  5.7850  6.6258  7.6290
  8.8468  10.3561  12.2759

>> u(11) = E2(1, 14.8, 3.45, 0.6, 0, 0, 0, 0, 2)

u =

  3.4500  3.9197  4.4548  5.0701  5.7850  6.6258  7.6290
  8.8468  10.3561  12.2759  14.8000

>> v(1) = E2(0, 14.8, 3.45, 0, 0.3, 0.3, 0.36, 85.6, 3)

v =

  3.4500

>> v(2) = E2(0.1, 14.8, 3.45, 0, 0.3, 0.3, 0.36, 85.6, 3)

v =

  3.4500  4.1564

>> v(3) = E2(0.2, 14.8, 3.45, 0, 0.3, 0.3, 0.36, 85.6, 3)
```

```
v =

  3.4500  4.1564  4.6041

>> v(4) = E2(0.3, 14.8, 3.45, 0, 0.3, 0.3, 0.36, 85.6, 3)

v =

  3.4500  4.1564  4.6041  5.0767

>> v(5) = E2(0.4, 14.8, 3.45, 0, 0.3, 0.3, 0.36, 85.6, 3)

v =

  3.4500  4.1564  4.6041  5.0767  5.6249

>> v(6) = E2(0.5, 14.8, 3.45, 0, 0.3, 0.3, 0.36, 85.6, 3)

v =

  3.4500  4.1564  4.6041  5.0767  5.6249  6.2878

>> v(7) = E2(0.6, 14.8, 3.45, 0, 0.3, 0.3, 0.36, 85.6, 3)

v =

  3.4500  4.1564  4.6041  5.0767  5.6249  6.2878  7.1155

>> v(8) = E2(0.7, 14.8, 3.45, 0, 0.3, 0.3, 0.36, 85.6, 3)

v =

  3.4500  4.1564  4.6041  5.0767  5.6249  6.2878  7.1155
  8.1845

>> v(9) = E2(0.8, 14.8, 3.45, 0, 0.3, 0.3, 0.36, 85.6, 3)

v =

  3.4500  4.1564  4.6041  5.0767  5.6249  6.2878  7.1155
  8.1845  9.6228

>> v(10) = E2(0.9, 14.8, 3.45, 0, 0.3, 0.3, 0.36, 85.6, 3)

v =

  3.4500  4.1564  4.6041  5.0767  5.6249  6.2878  7.1155
  8.1845  9.6228  11.6657
```

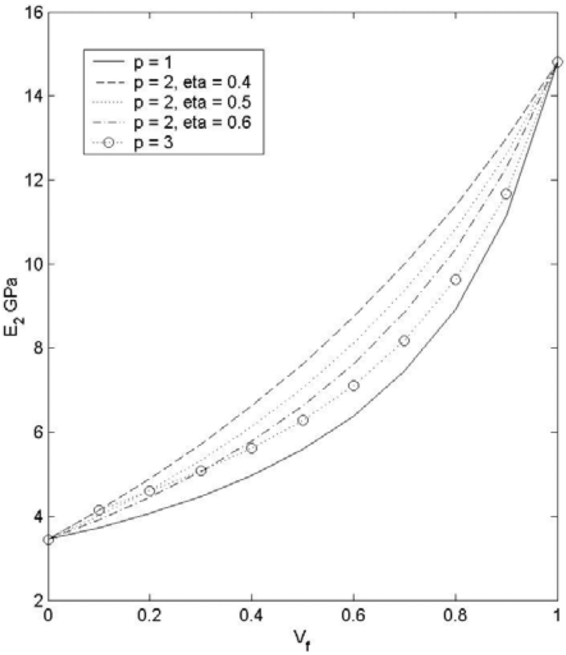

Fig. Variation of E_2 versus V^f for Problem 3.6

```
>> v(11) = E2(1, 14.8, 3.45, 0, 0.3, 0.3, 0.36, 85.6, 3)

v =

  3.4500  4.1564  4.6041  5.0767  5.6249  6.2878  7.1155
  8.1845  9.6228  11.6657  14.8000

>> x = [0 ; 0.1 ; 0.2 ; 0.3 ; 0.4 ; 0.5 ; 0.6 ; 0.7 ; 0.8 ;
       0.9 ; 1]

x =

        0
   0.1000
   0.2000
   0.3000
   0.4000
   0.5000
   0.6000
   0.7000
   0.8000
   0.9000
   1.0000
```

```
>> plot(x,y,'k-',x,z,'k--',x,w,'k:',x,u,'k-.',x,v,'ko:')
>> xlabel('V_f');
>> ylabel('E_2 GPa');
>> legend('p = 1', 'p = 2, eta = 0.4', 'p = 2, eta = 0.5',
   'p = 2, eta = 0.6', 'p = 3', 5);
```

Problem 3.7

The shear modulus G_{12} is calculated in GPa using three different formulas using the MATLAB function $G12$ as follows. Notice that the second and third values obtained are very close.

```
>> G12(0.55, 28.3, 1.27, 0, 1)

ans =

   2.6755

>> G12(0.55, 28.3, 1.27, 0.6, 2)

ans =

   3.5340

>> G12(0.55, 28.3, 1.27, 0, 3)

ans =

   3.8382
```

Problem 3.8

```
>> y(1) = G12(0, 28.3, 1.27, 0, 1)

y =

  1.2700

>> y(2) = G12(0.1, 28.3, 1.27, 0, 1)

y =

  1.2700  1.4041

>> y(3) = G12(0.2, 28.3, 1.27, 0, 1)
```

```
y =

  1.2700  1.4041  1.5699

>> y(4) = G12(0.3, 28.3, 1.27, 0, 1)

y =

  1.2700  1.4041  1.5699  1.7801

>> y(5) = G12(0.4, 28.3, 1.27, 0, 1)

y =

  1.2700  1.4041  1.5699  1.7801  2.0552

>> y(6) = G12(0.5, 28.3, 1.27, 0, 1)

y =

  1.2700  1.4041  1.5699  1.7801  2.0552  2.4309

>> y(7) = G12(0.6, 28.3, 1.27, 0, 1)

y =

  1.2700  1.4041  1.5699  1.7801  2.0552  2.4309  2.9748

>> y(8) = G12(0.7, 28.3, 1.27, 0, 1)

y =

  1.2700  1.4041  1.5699  1.7801  2.0552  2.4309  2.9748
  3.8321

>> y(9) = G12(0.8, 28.3, 1.27, 0, 1)

y =

  1.2700  1.4041  1.5699  1.7801  2.0552  2.4309  2.9748
  3.8321  5.3836

>> y(10) = G12(0.9, 28.3, 1.27, 0, 1)

y =

  1.2700  1.4041  1.5699  1.7801  2.0552  2.4309  2.9748
  3.8321  5.3836  9.0463
```

```
>> y(11) = G12(1, 28.3, 1.27, 0, 1)

y =

  Columns 1 through 10

    1.2700  1.4041  1.5699  1.7801  2.0552  2.4309  2.9748
    3.8321  5.3836  9.0463

  Column 11

    28.3000

>> z(1) = G12(0, 28.3, 1.27, 0.6, 2)

z =

  1.2700

>> z(2) = G12(0.1, 28.3, 1.27, 0.6, 2)

z =

  1.2700  1.4928

>> z(3) = G12(0.2, 28.3, 1.27, 0.6, 2)

z =

  1.2700  1.4928  1.7661

>> z(4) = G12(0.3, 28.3, 1.27, 0.6, 2)

z =

  1.2700  1.4928  1.7661  2.1095

>> z(5) = G12(0.4, 28.3, 1.27, 0.6, 2)

z =

  1.2700  1.4928  1.7661  2.1095  2.5538

>> z(6) = G12(0.5, 28.3, 1.27, 0.6, 2)

z =

  1.2700  1.4928  1.7661  2.1095  2.5538  3.1510
```

```
>> z(7) = G12(0.6, 28.3, 1.27, 0.6, 2)

z =

  1.2700  1.4928  1.7661  2.1095  2.5538  3.1510  3.9966

>> z(8) = G12(0.7, 28.3, 1.27, 0.6, 2)

z =

  1.2700  1.4928  1.7661  2.1095  2.5538  3.1510  3.9966
  5.2863

>> z(9) = G12(0.8, 28.3, 1.27, 0.6, 2)

z =

  1.2700  1.4928  1.7661  2.1095  2.5538  3.1510  3.9966
  5.2863  7.4945

>> z(10) = G12(0.9, 28.3, 1.27, 0.6, 2)

z =

  1.2700  1.4928  1.7661  2.1095  2.5538  3.1510  3.9966
  5.2863  7.4945  12.1448

>> z(11) = G12(1, 28.3, 1.27, 0.6, 2)

z =

  Columns 1 through 10

    1.2700  1.4928  1.7661  2.1095  2.5538  3.1510  3.9966
    5.2863  7.4945  12.1448

  Column 11

    28.3000

>> w(1) = G12(0, 28.3, 1.27, 0, 3)

w =

   1.2700
```

```
>> w(2) = G12(0.1, 28.3, 1.27, 0, 3)

w =

   1.2700  1.5255

>> w(3) = G12(0.2, 28.3, 1.27, 0, 3)

w =

   1.2700  1.5255  1.8383

>> w(4) = G12(0.3, 28.3, 1.27, 0, 3)

w =

   1.2700  1.5255  1.8383  2.2297

>> w(5) = G12(0.4, 28.3, 1.27, 0, 3)

w =

   1.2700  1.5255  1.8383  2.2297  2.7340

>> w(6) = G12(0.5, 28.3, 1.27, 0, 3)

w =

   1.2700  1.5255  1.8383  2.2297  2.7340  3.4082

>> w(7) = G12(0.6, 28.3, 1.27, 0, 3)

w =

   1.2700  1.5255  1.8383  2.2297  2.7340  3.4082  4.3552

>> w(8) = G12(0.7, 28.3, 1.27, 0, 3)

w =

   1.2700  1.5255  1.8383  2.2297  2.7340  3.4082  4.3552
   5.7830

>> w(9) = G12(0.8, 28.3, 1.27, 0, 3)
```

```
w =

   1.2700  1.5255  1.8383  2.2297  2.7340  3.4082  4.3552
   5.7830  8.1823

>> w(10) = G12(0.9, 28.3, 1.27, 0, 3)

w =

   1.2700  1.5255  1.8383  2.2297  2.7340  3.4082  4.3552
   5.7830  8.1823  13.0553

>> w(11) = G12(1, 28.3, 1.27, 0, 3)

w =

  Columns 1 through 10

    1.2700  1.5255  1.8383  2.2297  2.7340  3.4082  4.3552
    5.7830  8.1823  13.0553

  Column 11

    28.3000

>> x = [0 ; 0.1 ; 0.2 ; 0.3 ; 0.4 ; 0.5 ; 0.6 ; 0.7 ; 0.8 ;
      0.9 ; 1]

x =

        0
   0.1000
   0.2000
   0.3000
   0.4000
   0.5000
   0.6000
   0.7000
   0.8000
   0.9000
   1.0000

>> plot(x,y,‘k-’,x,z,‘k--’,x,w,‘k-.’)
>> xlabel(‘V ^ f’);
>> ylabel(‘G_{}12{} GPa’);
>> legend(‘p = 1’, ‘p = 2’, ‘p = 3’, 3);
```

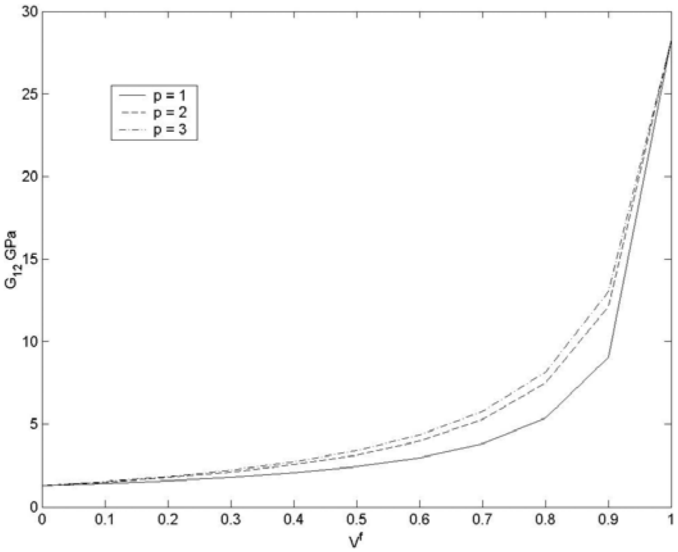

Fig. Variation of G_{12} versus V^f for Problem 3.8

Problem 3.9

First, the longitudinal coefficient of thermal expansion α_1 is calculated in /K as follows:

```
>> Alpha1(0.6, 233, 4.62, -0.540e-6, 41.4e-6)

ans =

  7.1671e-009
```

Next, the transverse coefficient of thermal expansion α_2 is calculated in /K using two different formulas as follows. Notice that in the second formula, we need to calculate also the value of the longitudinal modulus E_1. Note also that the two values obtained are comparable and very close to each other.

```
>> Alpha2(0.6, 10.10e-6, 41.4e-6, 0, 0, 0, 0, 0, 0, 1)

ans =

  2.2620e-005

>> E1 = E1(0.6, 233, 4.62)

E1 =

  141.6480
```

```
>> Alpha2(0.6, 10.10e-6, 41.4e-6, E1, 233, 4.62, 0.200,
         0.360, -0.540e-6, 2)

ans =

   2.8515e-005
```

Problem 3.10

$$E_1 = E^f V^f + E^m V^m + E^i V^i$$

Note that the derivation of the above equation is very similar to the derivation in Example 3.1.

Problem 4.1

$$\begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \gamma_{12} \end{Bmatrix} = \begin{bmatrix} \dfrac{1}{E_1} & \dfrac{-\nu_{12}}{E_1} & 0 \\ \dfrac{-\nu_{12}}{E_1} & \dfrac{1}{E_2} & 0 \\ 0 & 0 & \dfrac{1}{G_{12}} \end{bmatrix} \begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{Bmatrix}$$

Problem 4.2

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{Bmatrix} = \begin{bmatrix} \dfrac{E_1}{1-\nu_{12}\nu_{21}} & \dfrac{\nu_{12}E_2}{1-\nu_{12}\nu_{21}} & 0 \\ \dfrac{\nu_{12}E_2}{1-\nu_{12}\nu_{21}} & \dfrac{E_2}{1-\nu_{12}\nu_{21}} & 0 \\ 0 & 0 & G_{12} \end{bmatrix} \begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \gamma_{12} \end{Bmatrix}$$

where $\nu_{12}E_2 = \nu_{21}E_1$.

Problem 4.3

$$\begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \gamma_{12} \end{Bmatrix} = \begin{bmatrix} \dfrac{1}{E} & \dfrac{-\nu}{E} & 0 \\ \dfrac{-\nu}{E} & \dfrac{1}{E} & 0 \\ 0 & 0 & \dfrac{2(1+\nu)}{E} \end{bmatrix} \begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{Bmatrix}$$

Problem 4.4

$$\left\{\begin{array}{c}\sigma_1\\ \sigma_2\\ \tau_{12}\end{array}\right\} = \begin{bmatrix} \dfrac{E}{1-\nu^2} & \dfrac{\nu E}{1-\nu^2} & 0 \\ \dfrac{\nu E}{1-\nu^2} & \dfrac{E}{1-\nu^2} & 0 \\ 0 & 0 & \dfrac{E}{2(1+\nu)} \end{bmatrix} \left\{\begin{array}{c}\varepsilon_1\\ \varepsilon_2\\ \gamma_{12}\end{array}\right\}$$

Problem 4.5

```
>> S = ReducedCompliance(50.0, 15.2, 0.254, 4.70)

S =

   0.0200   -0.0051         0
  -0.0051    0.0658         0
        0         0    0.2128

>> Q = ReducedStiffness(50.0, 15.2, 0.254, 4.70)

Q =

   51.0003    3.9380         0
    3.9380   15.5041         0
         0         0    4.7000

>> S*Q

ans =

    1.0000         0         0
         0    1.0000         0
         0         0    1.0000
```

Problem 4.6

```
>> S = OrthotropicCompliance(155.0, 12.10, 12.10, 0.248, 0.458,
       0.248, 4.40, 3.20, 4.40)

S =

   0.0065   -0.0016   -0.0016         0         0         0
  -0.0016    0.0826   -0.0379         0         0         0
```

```
  -0.0016   -0.0379    0.0826         0         0         0
        0         0         0    0.3125         0         0
        0         0         0         0    0.2273         0
        0         0         0         0         0    0.2273

>> sigma1 = 0

sigma1 =

     0

>> sigma2 = -2.5/(200*0.200)

sigma2 =

   -0.0625

>> epsilon3 = S(1,3)*sigma1 + S(2,3)*sigma2

epsilon3 =

    0.0024
```

Problem 4.7

```
>> S = ReducedIsotropicCompliance(72.4, 0.3)

S =

   0.0138   -0.0041         0
  -0.0041    0.0138         0
        0         0    0.0359

>> Q = ReducedIsotropicStiffness(72.4, 0.3)

Q =

   79.5604   23.8681         0
   23.8681   79.5604         0
         0         0   27.8462

>> S*Q

ans =

   1.0000         0         0
        0    1.0000         0
        0         0    1.0000
```

Problem 4.8

```
>> S = OrthotropicCompliance(155.0, 12.10, 12.10, 0.248, 0.458,
       0.248, 4.40, 3.20, 4.40)

S =

    0.0065   -0.0016   -0.0016         0         0         0
   -0.0016    0.0826   -0.0379         0         0         0
   -0.0016   -0.0379    0.0826         0         0         0
         0         0         0    0.3125         0         0
         0         0         0         0    0.2273         0
         0         0         0         0         0    0.2273

>> sigma1 = 4/(200*0.200)

sigma1 =

    0.1000

>> sigma2 = 0

sigma2 =

    0

>> epsilon3 = S(1,3)*sigma1 + S(2,3)*sigma2

epsilon3 =

 -1.6000e-004
```

Problem 4.9

```
function y = ReducedStiffness2(E1,E2,NU12,G12)
%ReducedStiffness2  This function returns the reduced
%                   stiffness matrix for fiber-reinforced
%                   materials.
%                   There are four arguments representing
%                   four material constants.
%                   The size of the reduced compliance
%                   matrix is 3 x 3. The reuduced stiffness
%                   matrix is calculated as the inverse of
%                   the reduced compliance matrix.
z = [1/E1 -NU12/E1 0 ; -NU12/E1 1/E2 0 ; 0 0 1/G12];
y = inv(z);
```

```
function y = ReducedIsotropicStiffness2(E,NU)
%ReducedIsotropicStiffness2  This function returns the
%                            reduced isotropic stiffness
%                            matrix for fiber-reinforced
%                            materials.
%                            There are two arguments
%                            representing two material
%                            constants. The size of the
%                            reduced compliance matrix is
%                            3 x 3. The reduced stiffness
%                            matrix is calculated
%                            as the inverse of the reduced
%                            compliance matrix.
z = [1/E -NU/E 0 ; -NU/E 1/E 0 ; 0 0 2*(1+NU)/E];
y = inv(z);
```

Problem 4.10

$$\begin{Bmatrix} \varepsilon_1 - \alpha_1 \Delta T - \beta_1 \Delta M \\ \varepsilon_2 - \alpha_2 \Delta T - \beta_2 \Delta M \\ \gamma_{12} \end{Bmatrix} = \begin{bmatrix} S_{11} & S_{12} & 0 \\ S_{12} & S_{22} & 0 \\ 0 & 0 & S_{66} \end{bmatrix} \begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{Bmatrix}$$

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{Bmatrix} = \begin{bmatrix} Q_{11} & Q_{12} & 0 \\ Q_{12} & Q_{22} & 0 \\ 0 & 0 & Q_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_1 - \alpha_1 \Delta T - \beta_1 \Delta M \\ \varepsilon_2 - \alpha_2 \Delta T - \beta_2 \Delta M \\ \gamma_{12} \end{Bmatrix}$$

Problem 5.1

From an introductory course on mechanics of materials, we have the following stress transformation equations between the 1-2-3 coordinate system and the x-y-z global coordinate system:

$$\begin{aligned}
\sigma_1 &= \sigma_x \cos^2\theta + \sigma_y \sin^2\theta + 2\tau_{xy} \sin\theta\cos\theta \\
\sigma_2 &= \sigma_x \sin^2\theta + \sigma_y \cos^2\theta - 2\tau_{xy} \sin\theta\cos\theta \\
\sigma_3 &= \sigma_z \\
\tau_{23} &= \tau_{yz}\cos\theta - \tau_{xz}\sin\theta \\
\tau_{13} &= \tau_{yz}\sin\theta + \tau_{xz}\cos\theta \\
\tau_{12} &= -\sigma_x \sin\theta\cos\theta + \sigma_y \sin\theta\cos\theta + \tau_{xy}(\cos^2\theta - \sin^2\theta)
\end{aligned}$$

For the case of plane stress, we already have $\sigma_3 = \tau_{23} = \tau_{13} = 0$. Substitute this into the third, fourth, and fifth equations above and rearrange the terms to obtain:

$$\sigma_z = 0$$
$$\tau_{yz}\cos\theta - \tau_{xz}\sin\theta = 0$$
$$\tau_{yz}\sin\theta + \tau_{xz}\cos\theta = 0$$

It is clear now that $\sigma_z = 0$. Next, we solve the last two equations above by multiplying the first equation by $\cos\theta$ and the second equation by $\sin\theta$. Then, we add the two equation to obtain:

$$\tau_{yz}(\cos^2\theta + \sin^2\theta) = 0$$

However, we know that $\cos^2\theta + \sin^2\theta = 1$. Therefore, we conclude that $\tau_{yz} = 0$. It also follows immediately that $\tau_{xz} = 0$ also.

Problem 5.2

From an introductory course on mechanics of materials, we have the following stress transformation equations between the 1-2-3 coordinate system and the x-y-z global coordinate system:

$$\begin{aligned}
\sigma_1 &= \sigma_x\cos^2\theta + \sigma_y\sin^2\theta + 2\tau_{xy}\sin\theta\cos\theta \\
\sigma_2 &= \sigma_x\sin^2\theta + \sigma_y\cos^2\theta - 2\tau_{xy}\sin\theta\cos\theta \\
\tau_{12} &= -\sigma_x\sin\theta\cos\theta + \sigma_y\sin\theta\cos\theta + \tau_{xy}(\cos^2\theta - \sin^2\theta)
\end{aligned}$$

Write the above three equations in matrix form as follows:

$$\begin{Bmatrix}\sigma_1\\ \sigma_2\\ \tau_{12}\end{Bmatrix} = \begin{bmatrix}\cos^2\theta & \sin^2\theta & 2\sin\theta\cos\theta\\ \sin^2\theta & \cos^2\theta & -2\sin\theta\cos\theta\\ -\sin\theta\cos\theta & \sin\theta\cos\theta & \cos^2\theta - \sin^2\theta\end{bmatrix}\begin{Bmatrix}\sigma_x\\ \sigma_y\\ \tau_{xy}\end{Bmatrix}$$

Let $m = \cos\theta$ and $n = \sin\theta$. Therefore, we obtain the desired equation as follows:

$$\begin{Bmatrix}\sigma_1\\ \sigma_2\\ \tau_{12}\end{Bmatrix} = \begin{bmatrix}m^2 & n^2 & 2mn\\ n^2 & m^2 & -2mn\\ -mn & mn & m^2 - n^2\end{bmatrix}\begin{Bmatrix}\sigma_x\\ \sigma_y\\ \tau_{xy}\end{Bmatrix}$$

Problem 5.3

$$[T] = \begin{bmatrix}m^2 & n^2 & 2mn\\ n^2 & m^2 & -2mn\\ -mn & mn & m^2 - n^2\end{bmatrix}$$

Calculate the determinant of $[T]$ as follows:

$$
\begin{aligned}
|T| &= m^2 \begin{vmatrix} m^2 & -2mn \\ mn & m^2 - n^2 \end{vmatrix} - n^2 \begin{vmatrix} n^2 & -2mn \\ -mn & m^2 - n^2 \end{vmatrix} + 2mn \begin{vmatrix} n^2 & m^2 \\ -mn & mn \end{vmatrix} \\
&= \left(m^2 + n^2\right)^3 \\
&= 1
\end{aligned}
$$

The above is true since $m^2 + n^2 = \cos^2\theta + \sin^2\theta = 1$. Therefore, we obtain:

$$
\begin{aligned}
[T]^{-1} &= \frac{adj[T\}}{|T|} = adj[T] \\
&= \begin{bmatrix} m^2 & n^2 & -2mn \\ n^2 & m^2 & 2mn \\ mn & -mn & m^2 - n^2 \end{bmatrix}
\end{aligned}
$$

Problem 5.4

$$
\begin{aligned}
[\bar{S}] &= [T]^{-1}[S][T] \\
[\bar{S}]^{-1} &= \left([T]^{-1}[S][T]\right)^{-1} = [T]^{-1}[S]^{-1}\left([T]^{-1}\right)^{-1} = [T]^{-1}[Q][T] = [\bar{Q}]
\end{aligned}
$$

Similarly, we also have the other way:

$$
\begin{aligned}
[\bar{Q}] &= [T]^{-1}[Q][T] \\
[\bar{Q}]^{-1} &= \left([T]^{-1}[Q][T]\right)^{-1} = [T]^{-1}[Q]^{-1}\left([T]^{-1}\right)^{-1} = [T]^{-1}[S][T] = [\bar{S}]
\end{aligned}
$$

Problem 5.5

Multiply the three matrices in (5.13) in book as follows:

$$
\begin{bmatrix} \bar{Q}_{11} & \bar{Q}_{12} & \bar{Q}_{16} \\ \bar{Q}_{12} & \bar{Q}_{22} & \bar{Q}_{26} \\ \bar{Q}_{16} & \bar{Q}_{26} & \bar{Q}_{66} \end{bmatrix} = \begin{bmatrix} m^2 & n^2 & -2mn \\ n^2 & m^2 & 2mn \\ mn & -mn & m^2 - n^2 \end{bmatrix} \begin{bmatrix} Q_{11} & Q_{12} & 0 \\ Q_{12} & Q_{22} & 0 \\ 0 & 0 & Q_{66} \end{bmatrix}
$$
$$
\begin{bmatrix} m^2 & n^2 & 2mn \\ n^2 & m^2 & -2mn \\ -mn & mn & m^2 - n^2 \end{bmatrix}
$$

The above multiplication can be performed either manually or using a computer algebra system like MAPLE or MATHEMATICA or the MATLAB Symbolic Math Toolbox. Therefore, we obtain the following expression:

$$\begin{aligned}
\bar{Q}_{11} &= Q_{11}m^4 + 2(Q_{12} + 2Q_{66})n^2m^2 + Q_{22}n^4 \\
\bar{Q}_{12} &= (Q_{11} + Q_{22} - 4Q_{66})n^2m^2 + Q_{12}(n^4 + m^4) \\
\bar{Q}_{16} &= (Q_{11} - Q_{12} - 2Q_{66})nm^3 + (Q_{12} - Q_{22} + 2Q_{66})n^3m \\
\bar{Q}_{22} &= Q_{11}n^4 + 2(Q_{12} + 2Q_{66})n^2m^2 + Q_{22}m^4 \\
\bar{Q}_{26} &= (Q_{11} - Q_{12} - 2Q_{66})n^3m + (Q_{12} - Q_{22} + 2Q_{66})nm^3 \\
\bar{Q}_{66} &= (Q_{11} + Q_{22} - 2Q_{12} - 2Q_{66})n^2m^2 + Q_{66}(n^4 + m^4)
\end{aligned}$$

Problem 5.6

```
function y = Tinv2(theta)
%Tinv2  This function returns the inverse of the
%       transformation matrix T
%       given the orientation angle "theta".
%       There is only one argument representing "theta"
%       The size of the matrix is 3 x 3.
%       The angle "theta" must be given in degrees.
m = cos(theta*pi/180);
n = sin(theta*pi/180);
x = [m*m n*n 2*m*n ; n*n m*m -2*m*n ; -m*n m*n m*m-n*n];
y = inv(x);
```

Problem 5.7

```
function y = Sbar2(S,T)
%Sbar2  This function returns the transformed reduced
%       compliance matrix "Sbar" given the reduced
%       compliance matrix S and the transformation
%       matrix T.
%       There are two arguments representing S and T
%       The size of the matrix is 3 x 3.
Tinv = inv(T);
y = Tinv*S*T;
```

```
function y = Qbar2(Q,T)
%Qbar2  This function returns the transformed reduced
%       stiffness matrix "Qbar" given the reduced
%       stiffness matrix Q and the transformation
%       matrix T.
%       There are two arguments representing Q and T
```

```
%       The size of the matrix is 3 x 3.
Tinv = inv(T);
y = Tinv*Q*T;
```

Problem 5.8

```
>> S = ReducedCompliance(50.0, 15.20, 0.254, 4.70)

S =

    0.0200   -0.0051         0
   -0.0051    0.0658         0
         0         0    0.2128

>> S1 = Sbar(S, -90)

S1 =

    0.0658   -0.0051   -0.0000
   -0.0051    0.0200    0.0000
   -0.0000    0.0000    0.2128

>> S2 = Sbar(S, -80)

S2 =

    0.0740   -0.0147   -0.0451
   -0.0147    0.0310    0.0608
   -0.0226    0.0304    0.1935

>> S3 = Sbar(S, -70)

S3 =

    0.0945   -0.0391   -0.0664
   -0.0391    0.0594    0.0959
   -0.0332    0.0479    0.1447

>> S4 = Sbar(S, -60)

S4 =

    0.1161   -0.0669   -0.0515
   -0.0669    0.0932    0.0912
   -0.0258    0.0456    0.0892
```

```
>> S5 = Sbar(S, -50)

S5 =

    0.1268   -0.0850   -0.0056
   -0.0850    0.1188    0.0507
   -0.0028    0.0254    0.0529

>> S6 = Sbar(S, -40)

S6 =

    0.1188   -0.0850    0.0507
   -0.0850    0.1268   -0.0056
    0.0254   -0.0028    0.0529

>> S7 = Sbar(S, -30)

S7 =

    0.0932   -0.0669    0.0912
   -0.0669    0.1161   -0.0515
    0.0456   -0.0258    0.0892

>> S8 = Sbar(S, -20)

S8 =

    0.0594   -0.0391    0.0959
   -0.0391    0.0945   -0.0664
    0.0479   -0.0332    0.1447

>> S9 = Sbar(S, -10)

S9 =

    0.0310   -0.0147    0.0608
   -0.0147    0.0740   -0.0451
    0.0304   -0.0226    0.1935
```


```
>> S9 = Sbar(S, -10)

S9 =

    0.0310   -0.0147    0.0608
   -0.0147    0.0740   -0.0451
    0.0304   -0.0226    0.1935
```

```
>> S10 = Sbar(S, 0)

S10 =

    0.0200   -0.0051         0
   -0.0051    0.0658         0
         0         0    0.2128

>> S11 = Sbar(S, 10)

S11 =

    0.0310   -0.0147   -0.0608
   -0.0147    0.0740    0.0451
   -0.0304    0.0226    0.1935

>> S12 = Sbar(S, 20)

S12 =

    0.0594   -0.0391   -0.0959
   -0.0391    0.0945    0.0664
   -0.0479    0.0332    0.1447

>> S13 = Sbar(S, 30)

S13 =

    0.0932   -0.0669   -0.0912
   -0.0669    0.1161    0.0515
   -0.0456    0.0258    0.0892

>> S14 = Sbar(S, 40)

S14 =

    0.1188   -0.0850   -0.0507
   -0.0850    0.1268    0.0056
   -0.0254    0.0028    0.0529

>> S15 = Sbar(S, 50)

S15 =

    0.1268   -0.0850    0.0056
   -0.0850    0.1188   -0.0507
    0.0028   -0.0254    0.0529
```

```
>> S16 = Sbar(S, 60)

S16 =

    0.1161   -0.0669    0.0515
   -0.0669    0.0932   -0.0912
    0.0258   -0.0456    0.0892

>> S17 = Sbar(S, 70)

S17 =

    0.0945   -0.0391    0.0664
   -0.0391    0.0594   -0.0959
    0.0332   -0.0479    0.1447

>> S18 = Sbar(S, 80)

S18 =

    0.0740   -0.0147    0.0451
   -0.0147    0.0310   -0.0608
    0.0226   -0.0304    0.1935

>> S19 = Sbar(S, 90)

S19 =

    0.0658   -0.0051    0.0000
   -0.0051    0.0200   -0.0000
    0.0000   -0.0000    0.2128

>> x = [-90 -80 -70 -60 -50 -40 -30 -20 -10 0 10 20 30 40 50
       60 70 80 90]

x =

   -90   -80   -70   -60   -50   -40   -30   -20   -10
    0    10    20    30    40    50    60    70    80    90

>> y1 = [S1(1,1) S2(1,1) S3(1,1) S4(1,1) S5(1,1) S6(1,1)
        S7(1,1) S8(1,1) S9(1,1) S10(1,1) S11(1,1) S12(1,1) S13(1,1)
        S14(1,1) S15(1,1) S16(1,1) S17(1,1) S18(1,1) S19(1,1)]

y1 =

  Columns 1 through 14

     0.0658    0.0740    0.0945    0.1161    0.1268    0.1188
```

```
    0.0932    0.0594    0.0310    0.0200    0.0310    0.0594
    0.0932    0.1188

  Columns 15 through 19

    0.1268    0.1161    0.0945    0.0740    0.0658

>> plot(x,y1)
>> xlabel('\theta (degrees)');
>> ylabel('S^{-}_{11} (GPa)^{-1}');
```

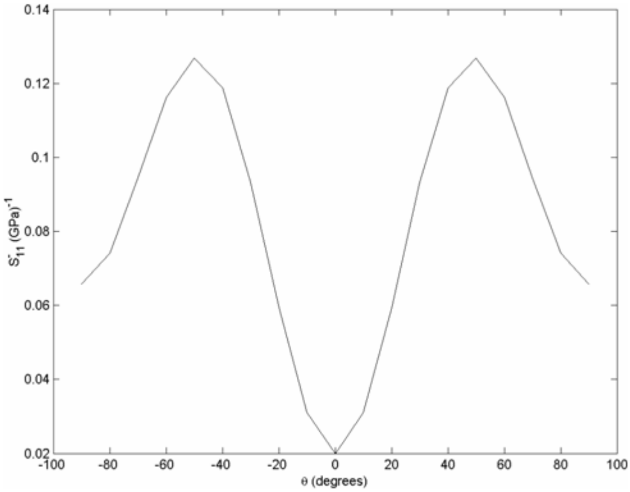

Fig. Variation of $\bar{S}_{11}$ versus θ for Problem 5.8

```
>> y2 = [S1(1,2) S2(1,2) S3(1,2) S4(1,2) S5(1,2) S6(1,2) S7(1,2)
        S8(1,2) S9(1,2) S10(1,2) S11(1,2) S12(1,2) S13(1,2) S14(1,2)
        S15(1,2) S16(1,2) S17(1,2) S18(1,2) S19(1,2)]

y2 =

  Columns 1 through 14

   -0.0051   -0.0147   -0.0391   -0.0669   -0.0850   -0.0850
   -0.0669   -0.0391   -0.0147   -0.0051   -0.0147   -0.0391
   -0.0669   -0.0850

  Columns 15 through 19

   -0.0850   -0.0669   -0.0391   -0.0147   -0.0051
```

```
>> plot(x,y2)
>> xlabel('\theta (degrees)');
>> ylabel('S^{-}_{12} (GPa)^{-1}');
```

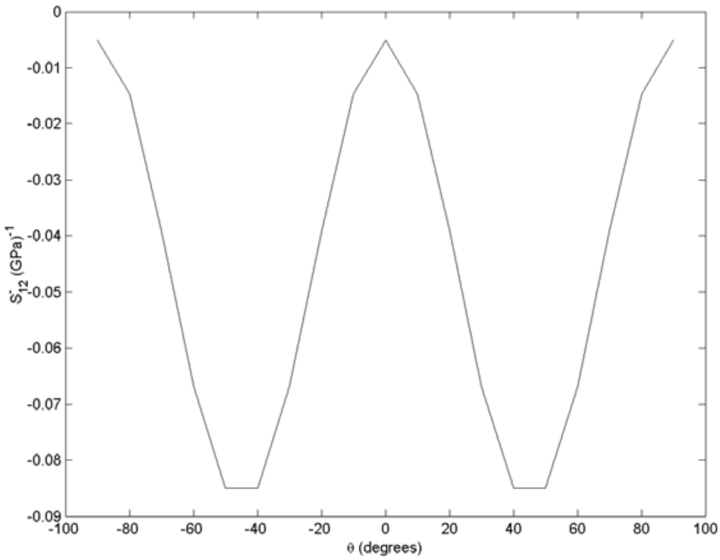

Fig. Variation of $\bar{S}_{12}$ versus θ for Problem 5.8

```
>> y3 = [S1(1,3) S2(1,3) S3(1,3) S4(1,3) S5(1,3) S6(1,3) S7(1,3)
        S8(1,3) S9(1,3) S10(1,3) S11(1,3) S12(1,3) S13(1,3) S14(1,3)
        S15(1,3) S16(1,3) S17(1,3) S18(1,3) S19(1,3)]

y3 =

  Columns 1 through 14

    -0.0000   -0.0451   -0.0664   -0.0515   -0.0056    0.0507
     0.0912    0.0959    0.0608         0   -0.0608   -0.0959
    -0.0912   -0.0507

  Columns 15 through 19

     0.0056    0.0515    0.0664    0.0451    0.0000

>> plot(x,y3)
>> xlabel('\theta (degrees)');
>> ylabel('S^{-}_{16} (GPa)^{-1}');
```

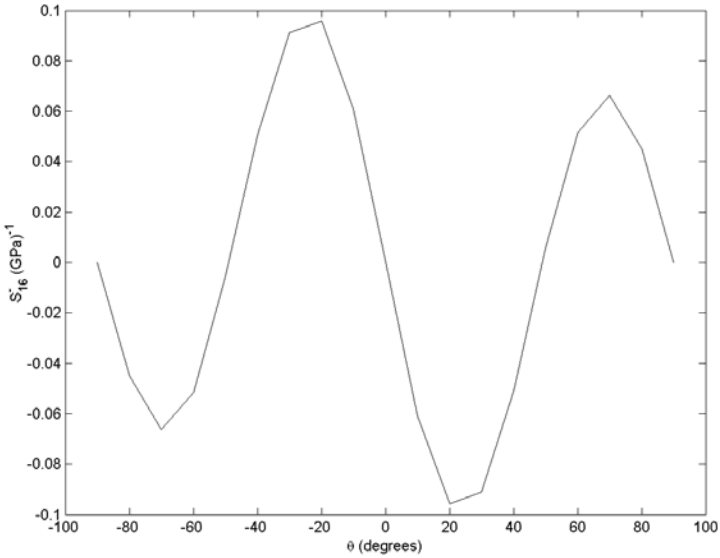

Fig. Variation of $\bar{S}_{16}$ versus θ for Problem 5.8

```
>> y4 = [S1(2,2) S2(2,2) S3(2,2) S4(2,2) S5(2,2) S6(2,2) S7(2,2)
        S8(2,2) S9(2,2) S10(2,2) S11(2,2) S12(2,2) S13(2,2) S14(2,2)
        S15(2,2) S16(2,2) S17(2,2) S18(2,2) S19(2,2)]

y4 =

  Columns 1 through 14

    0.0200    0.0310    0.0594    0.0932    0.1188    0.1268
    0.1161    0.0945    0.0740    0.0658    0.0740    0.0945
    0.1161    0.1268

  Columns 15 through 19

    0.1188    0.0932    0.0594    0.0310    0.0200

>> plot(x,y4)
>> xlabel('\theta (degrees)');
>> ylabel('S^{-}_{22} (GPa)^{-1}');
```

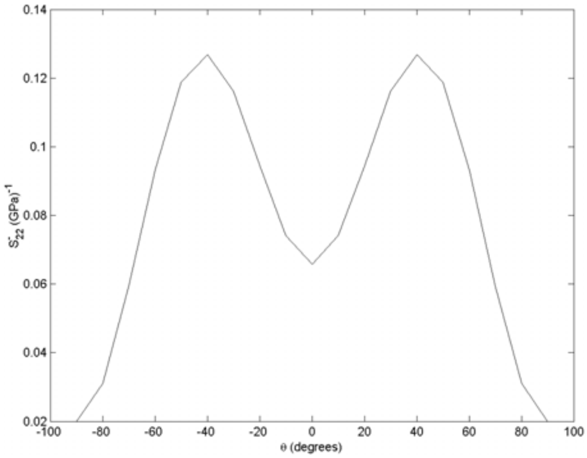

Fig. Variation of $\bar{S}_{22}$ versus θ for Problem 5.8

```
>> y5 = [S1(2,3) S2(2,3) S3(2,3) S4(2,3) S5(2,3) S6(2,3) S7(2,3)
        S8(2,3) S9(2,3) S10(2,3) S11(2,3) S12(2,3) S13(2,3) S14(2,3)
        S15(2,3) S16(2,3) S17(2,3) S18(2,3) S19(2,3)]
y5 =

  Columns 1 through 14

     0.0000    0.0608    0.0959    0.0912    0.0507   -0.0056
    -0.0515   -0.0664   -0.0451         0    0.0451    0.0664
     0.0515    0.0056

  Columns 15 through 19

    -0.0507   -0.0912   -0.0959   -0.0608   -0.0000

>> plot(x,y5)
>> xlabel('\theta (degrees)');
>> ylabel('S^{-}_{26} (GPa)^{-1}');

>> y6 = [S1(3,3) S2(3,3) S3(3,3) S4(3,3) S5(3,3) S6(3,3) S7(3,3)
        S8(3,3) S9(3,3) S10(3,3) S11(3,3) S12(3,3) S13(3,3) S14(3,3)
        S15(3,3) S16(3,3) S17(3,3) S18(3,3) S19(3,3)]

y6 =

  Columns 1 through 14

    0.2128    0.1935    0.1447    0.0892    0.0529    0.0529
    0.0892    0.1447    0.1935    0.2128    0.1935    0.1447
    0.0892    0.0529
```

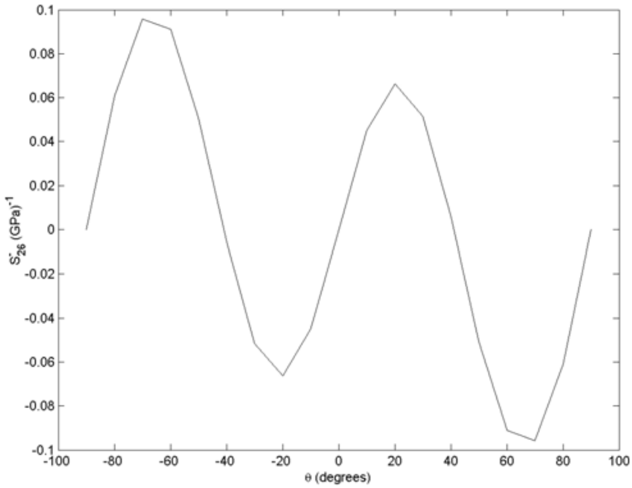

Fig. Variation of $\bar{S}_{26}$ versus θ for Problem 5.8

```
  Columns 15 through 19

    0.0529    0.0892    0.1447    0.1935    0.2128

>> plot(x,y6)
>> xlabel('\theta (degrees)');
>> ylabel('S^{-}_{66} (GPa)^{-1}');
```

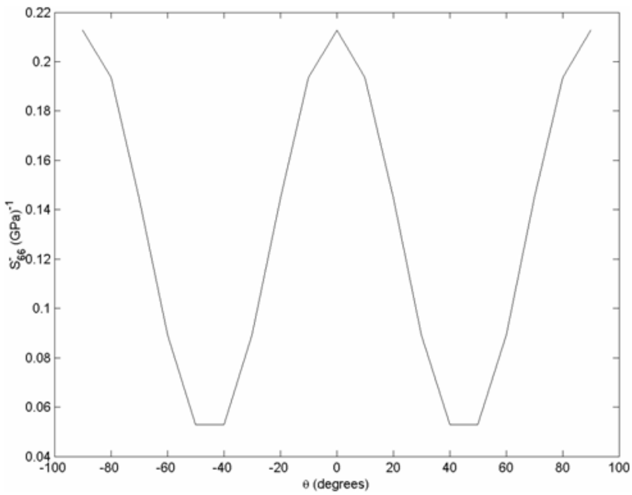

Fig. Variation of $\bar{S}_{66}$ versus θ for Problem 5.8

Problem 5.9

```
>> Q = ReducedStiffness(155.0, 12.10, 0.248, 4.40)

Q =

  155.7478    3.0153         0
    3.0153   12.1584         0
         0         0    4.4000

>> Q1 = Qbar(Q, -90)

Q1 =

   12.1584    3.0153    0.0000
    3.0153  155.7478   -0.0000
    0.0000   -0.0000    4.4000

>> Q2 = Qbar(Q, -80)

Q2 =

   12.0115    7.4919    0.0435
    7.4919  146.9414  -49.1540
    0.0218  -24.5770   13.3532

>> Q3 = Qbar(Q, -70)

Q3 =

   13.1434   18.8271   -8.4612
   18.8271  123.1392  -83.8363
   -4.2306  -41.9181   36.0236

>> Q4 = Qbar(Q, -60)

Q4 =

   19.3541   31.7170  -29.0342
   31.7170   91.1488  -95.3179
  -14.5171  -47.6589   61.8034

>> Q5 = Qbar(Q, -50)

Q5 =

   34.3711   40.1302  -57.6152
   40.1302   59.3051  -83.7927
  -28.8076  -41.8964   78.6299
```

```
>> Q6 = Qbar(Q, -40)

Q6 =

   59.3051   40.1302  -83.7927
   40.1302   34.3711  -57.6152
  -41.8964  -28.8076   78.6299

>> Q7 = Qbar(Q, -30)

Q7 =

   91.1488   31.7170  -95.3179
   31.7170   19.3541  -29.0342
  -47.6589  -14.5171   61.8034

>> Q8 = Qbar(Q, -20)

Q8 =

  123.1392   18.8271  -83.8363
   18.8271   13.1434   -8.4612
  -41.9181   -4.2306   36.0236

>> Q9 = Qbar(Q, -10)

Q9 =

  146.9414    7.4919  -49.1540
    7.4919   12.0115    0.0435
  -24.5770    0.0218   13.3532

>> Q10 = Qbar(Q, 0)

Q10 =

  155.7478    3.0153         0
    3.0153   12.1584         0
         0         0    4.4000

>> Q11 = Qbar(Q, 10)

Q11 =

  146.9414    7.4919   49.1540
    7.4919   12.0115   -0.0435
   24.5770   -0.0218   13.3532
```

```
>> Q12 = Qbar(Q, 20)

Q12 =

  123.1392   18.8271   83.8363
   18.8271   13.1434    8.4612
   41.9181    4.2306   36.0236

>> Q13 = Qbar(Q, 30)

Q13 =

   91.1488   31.7170   95.3179
   31.7170   19.3541   29.0342
   47.6589   14.5171   61.8034

>> Q14 = Qbar(Q, 40)

Q14 =

   59.3051   40.1302   83.7927
   40.1302   34.3711   57.6152
   41.8964   28.8076   78.6299

>> Q15 = Qbar(Q, 50)

Q15 =

   34.3711   40.1302   57.6152
   40.1302   59.3051   83.7927
   28.8076   41.8964   78.6299

>> Q16 = Qbar(Q, 60)

Q16 =

   19.3541   31.7170   29.0342
   31.7170   91.1488   95.3179
   14.5171   47.6589   61.8034


>> Q17 = Qbar(Q, 70)

Q17 =

   13.1434   18.8271    8.4612
   18.8271  123.1392   83.8363
    4.2306   41.9181   36.0236
```

```
>> Q18 = Qbar(Q, 80)

Q18 =

   12.0115    7.4919   -0.0435
    7.4919  146.9414   49.1540
   -0.0218   24.5770   13.3532

>> Q19 = Qbar(Q, 90)

Q19 =

   12.1584    3.0153   -0.0000
    3.0153  155.7478    0.0000
   -0.0000    0.0000    4.4000

>> x = [-90 -80 -70 -60 -50 -40 -30 -20 -10 0 10 20 30 40 50 60
       70 80 90]

x =

   -90   -80   -70   -60   -50   -40   -30   -20   -10     0    10
   20    30    40    50    60    70    80    90

>> y1 = [Q1(1,1) Q2(1,1) Q3(1,1) Q4(1,1) Q5(1,1) Q6(1,1) Q7(1,1)
        Q8(1,1) Q9(1,1) Q10(1,1) Q11(1,1) Q12(1,1) Q13(1,1) Q14(1,1)
        Q15(1,1) Q16(1,1) Q17(1,1) Q18(1,1) Q19(1,1)]

y1 =

  Columns 1 through 14

    12.1584   12.0115   13.1434   19.3541   34.3711   59.3051
    91.1488  123.1392  146.9414  155.7478  146.9414  123.1392
    91.1488   59.3051

  Columns 15 through 19

    34.3711   19.3541   13.1434   12.0115   12.1584


>> plot(x,y1)
>> xlabel('\theta (degrees)');
>> ylabel('Q^{-}_{11} (GPa)');
```

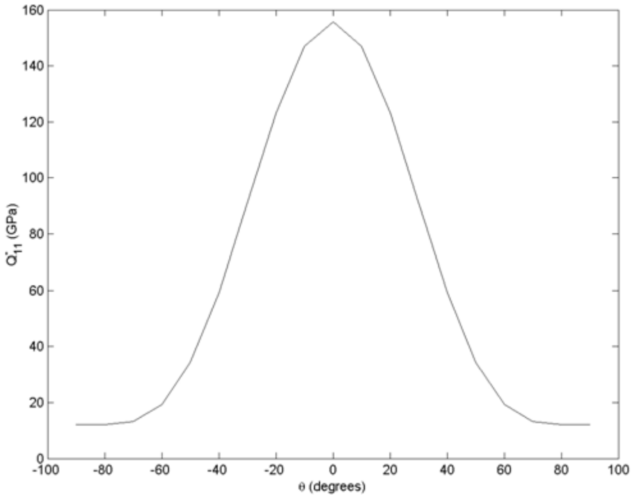

Fig. Variation of $\bar{Q}_{11}$ versus θ for Problem 5.9

```
>> y2 = [Q1(1,2) Q2(1,2) Q3(1,2) Q4(1,2) Q5(1,2) Q6(1,2) Q7(1,2)
        Q8(1,2) Q9(1,2) Q10(1,2) Q11(1,2) Q12(1,2) Q13(1,2) Q14(1,2)
        Q15(1,2) Q16(1,2) Q17(1,2) Q18(1,2) Q19(1,2)]

y2 =

  Columns 1 through 14

    3.0153    7.4919   18.8271   31.7170   40.1302   40.1302
    31.7170   18.8271    7.4919    3.0153    7.4919   18.8271
    31.7170   40.1302

  Columns 15 through 19

    40.1302   31.7170   18.8271    7.4919    3.0153

>> plot(x,y2)
>> xlabel('\theta (degrees)');
>> ylabel('Q^{-}_{12} (GPa)');
```

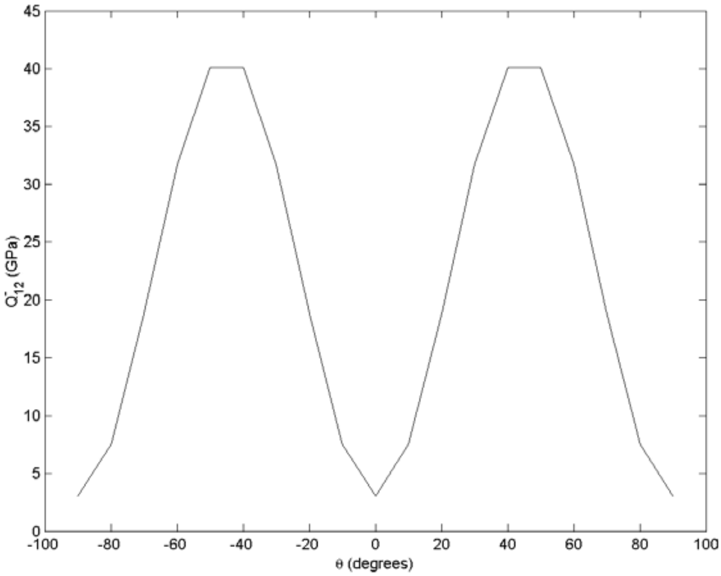

Fig. Variation of $\bar{Q}_{12}$ versus θ for Problem 5.9

```
>> y3 = [Q1(1,3) Q2(1,3) Q3(1,3) Q4(1,3) Q5(1,3) Q6(1,3) Q7(1,3)
        Q8(1,3) Q9(1,3) Q10(1,3) Q11(1,3) Q12(1,3) Q13(1,3) Q14(1,3)
        Q15(1,3) Q16(1,3) Q17(1,3) Q18(1,3) Q19(1,3)]

y3 =

  Columns 1 through 14

     0.0000    0.0435    -8.4612    -29.0342    -57.6152    -83.7927
    -95.3179    -83.8363    -49.1540    0    49.1540    83.8363
     95.3179    83.7927

  Columns 15 through 19

    57.6152    29.0342    8.4612    -0.0435    -0.0000

>> plot(x,y3)
>> xlabel('\theta (degrees)');
>> ylabel('Q^{-}_{16} (GPa)');
```

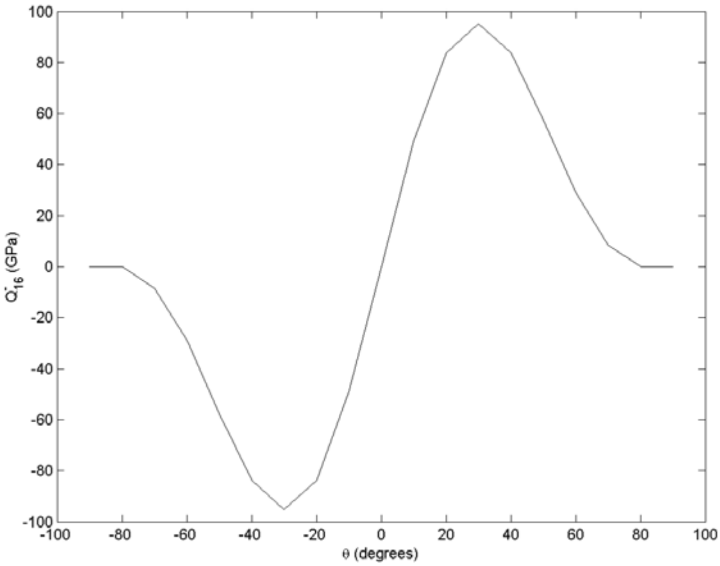

Fig. Variation of $\bar{Q}_{16}$ versus θ for Problem 5.9

```
>> y4 = [Q1(2,2) Q2(2,2) Q3(2,2) Q4(2,2) Q5(2,2) Q6(2,2) Q7(2,2)
        Q8(2,2) Q9(2,2) Q10(2,2) Q11(2,2) Q12(2,2) Q13(2,2) Q14(2,2)
        Q15(2,2) Q16(2,2) Q17(2,2) Q18(2,2) Q19(2,2)]

y4 =

  Columns 1 through 14

    155.7478  146.9414  123.1392   91.1488   59.3051   34.3711
    19.3541   13.1434   12.0115   12.1584   12.0115   13.1434   19.3541
    34.3711

  Columns 15 through 19

    59.3051   91.1488  123.1392  146.9414  155.7478

>> plot(x,y4)
>> xlabel('\theta (degrees)');
>> ylabel('Q^{-}_{22} (GPa)');
```

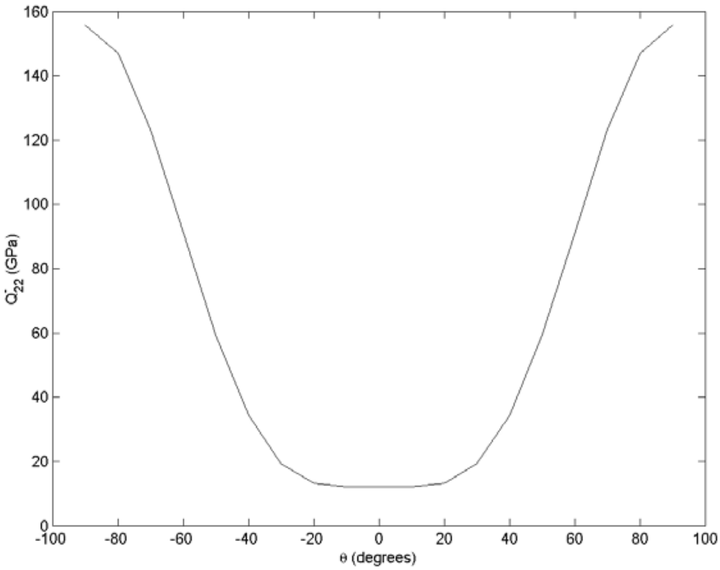

Fig. Variation of $\bar{Q}_{22}$ versus θ for Problem 5.9

```
>> y5 = [Q1(2,3) Q2(2,3) Q3(2,3) Q4(2,3) Q5(2,3) Q6(2,3) Q7(2,3)
        Q8(2,3) Q9(2,3) Q10(2,3) Q11(2,3) Q12(2,3) Q13(2,3) Q14(2,3)
        Q15(2,3) Q16(2,3) Q17(2,3) Q18(2,3) Q19(2,3)]

y5 =

  Columns 1 through 14

    -0.0000    -49.1540    -83.8363    -95.3179    -83.7927    -57.6152
    -29.0342    -8.4612    0.0435    0    -0.0435    8.4612
     29.0342    57.6152

  Columns 15 through 19

    83.7927    95.3179    83.8363    49.1540     0.0000

>> plot(x,y5)
>> xlabel('\theta (degrees)');
>> ylabel('Q^{-}_{26} (GPa)');
```

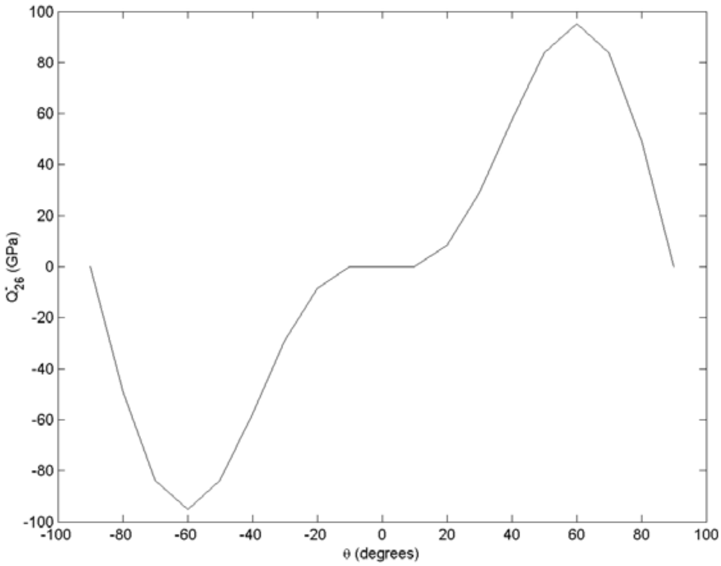

Fig. Variation of $\bar{Q}_{26}$ versus θ for Problem 5.9

```
>> y6 = [Q1(3,3) Q2(3,3) Q3(3,3) Q4(3,3) Q5(3,3) Q6(3,3) Q7(3,3)
        Q8(3,3) Q9(3,3) Q10(3,3) Q11(3,3) Q12(3,3) Q13(3,3) Q14(3,3)
        Q15(3,3) Q16(3,3) Q17(3,3) Q18(3,3) Q19(3,3)]

y6 =

  Columns 1 through 14

    4.4000   13.3532   36.0236   61.8034   78.6299   78.6299
    61.8034   36.0236   13.3532    4.4000   13.3532   36.0236
    61.8034   78.6299

  Columns 15 through 19

    78.6299   61.8034   36.0236   13.3532    4.4000

>> plot(x,y6)
>> xlabel('\theta (degrees)');
>> ylabel('Q^{-}_{66} (GPa)');
```

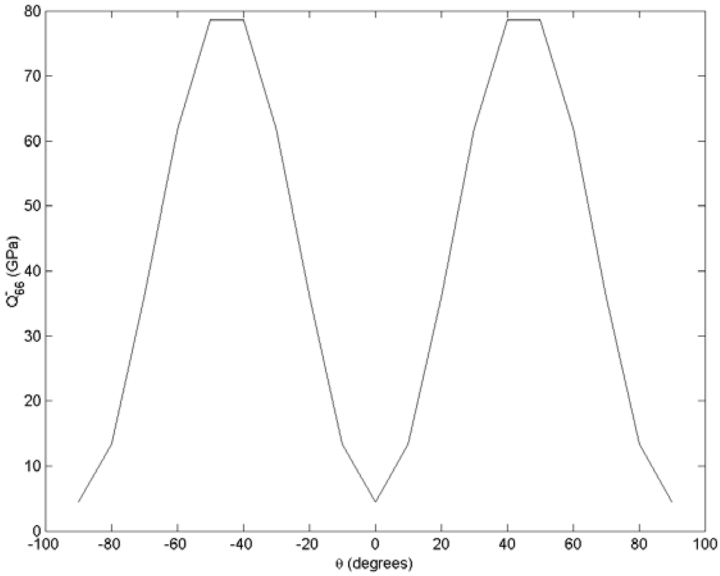

Fig. Variation of $\bar{Q}_{66}$ versus θ for Problem 5.9

Problem 5.10

```
>> Q = ReducedStiffness(50.0, 15.20, 0.254, 4.70)

Q =

   51.0003    3.9380         0
    3.9380   15.5041         0
         0         0    4.7000
>> Q1 = Qbar(Q, -90)

Q1 =

   15.5041    3.9380    0.0000
    3.9380   51.0003   -0.0000
    0.0000   -0.0000    4.7000

>> Q2 = Qbar(Q, -80)

Q2 =

   15.1348    5.3777    1.8406
    5.3777   48.4903  -13.9810
    0.9203   -6.9905    7.5793
```

```
>> Q3 = Qbar(Q, -70)

Q3 =

   14.5714    9.0230    0.7118
    9.0230   41.7630  -23.5283
    0.3559  -11.7642   14.8700

>> Q4 = Qbar(Q, -60)

Q4 =

   15.1478   13.1683   -4.7121
   13.1683   32.8959  -26.0285
   -2.3560  -13.0143   23.1606

>> Q5 = Qbar(Q, -50)

Q5 =

   18.2343   15.8740  -13.2692
   15.8740   24.3981  -21.6877
   -6.6346  -10.8439   28.5719

>> Q6 = Qbar(Q, -40)

Q6 =

   24.3981   15.8740  -21.6877
   15.8740   18.2343  -13.2692
  -10.8439   -6.6346   28.5719


>> Q7 = Qbar(Q, -30)

Q7 =

   32.8959   13.1683  -26.0285
   13.1683   15.1478   -4.7121
  -13.0143   -2.3560   23.1606

>> Q8 = Qbar(Q, -20)

Q8 =

   41.7630    9.0230  -23.5283
    9.0230   14.5714    0.7118
  -11.7642    0.3559   14.8700
```

```
>> Q9 = Qbar(Q, -10)

Q9 =

   48.4903    5.3777  -13.9810
    5.3777   15.1348    1.8406
   -6.9905    0.9203    7.5793

>> Q10 = Qbar(Q, 0)

Q10 =

   51.0003    3.9380         0
    3.9380   15.5041         0
         0         0    4.7000

>> Q11 = Qbar(Q, 10)

Q11 =

   48.4903    5.3777   13.9810
    5.3777   15.1348   -1.8406
    6.9905   -0.9203    7.5793

>> Q12 = Qbar(Q, 20)

Q12 =

   41.7630    9.0230   23.5283
    9.0230   14.5714   -0.7118
   11.7642   -0.3559   14.8700


>> Q13 = Qbar(Q, 30)

Q13 =

   32.8959   13.1683   26.0285
   13.1683   15.1478    4.7121
   13.0143    2.3560   23.1606

>> Q14 = Qbar(Q, 40)

Q14 =

   24.3981   15.8740   21.6877
   15.8740   18.2343   13.2692
   10.8439    6.6346   28.5719
```

```
>> Q15 = Qbar(Q, 50)

Q15 =

   18.2343   15.8740   13.2692
   15.8740   24.3981   21.6877
    6.6346   10.8439   28.5719

>> Q16 = Qbar(Q, 60)

Q16 =

   15.1478   13.1683    4.7121
   13.1683   32.8959   26.0285
    2.3560   13.0143   23.1606

>> Q17 = Qbar(Q, 70)

Q17 =

   14.5714    9.0230   -0.7118
    9.0230   41.7630   23.5283
   -0.3559   11.7642   14.8700

>> Q18 = Qbar(Q, 80)

Q18 =

   15.1348    5.3777   -1.8406
    5.3777   48.4903   13.9810
   -0.9203    6.9905    7.5793


>> Q19 = Qbar(Q, 90)

Q19 =

   15.5041    3.9380   -0.0000
    3.9380   51.0003    0.0000
   -0.0000    0.0000    4.7000

>> x = [-90 -80 -70 -60 -50 -40 -30 -20 -10 0 10 20 30 40 50 60 70
       80 90]

x =

   -90   -80   -70   -60   -50   -40   -30   -20   -10     0    10
   20    30    40    50    60    70    80    90
```

```
>> y1 = [Q1(1,1) Q2(1,1) Q3(1,1) Q4(1,1) Q5(1,1) Q6(1,1) Q7(1,1)
        Q8(1,1) Q9(1,1) Q10(1,1) Q11(1,1) Q12(1,1) Q13(1,1) Q14(1,1)
        Q15(1,1) Q16(1,1) Q17(1,1) Q18(1,1) Q19(1,1)]

y1 =

  Columns 1 through 14

    15.5041   15.1348   14.5714   15.1478   18.2343   24.3981
    32.8959   41.7630   48.4903   51.0003   48.4903   41.7630
    32.8959   24.3981

  Columns 15 through 19

    18.2343   15.1478   14.5714   15.1348   15.5041

>> plot(x,y1)
>> xlabel('\theta (degrees)');
>> ylabel('Q^{-}_{11} (GPa)');
```

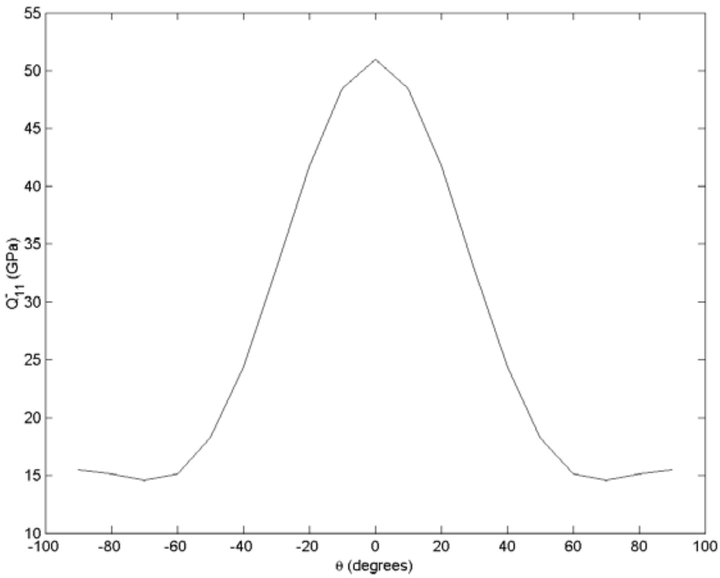

Fig. Variation of $\bar{Q}_{11}$ versus θ for Problem 5.10

```
>> y2 = [Q1(1,2) Q2(1,2) Q3(1,2) Q4(1,2) Q5(1,2) Q6(1,2) Q7(1,2)
        Q8(1,2) Q9(1,2) Q10(1,2) Q11(1,2) Q12(1,2) Q13(1,2) Q14(1,2)
        Q15(1,2) Q16(1,2) Q17(1,2) Q18(1,2) Q19(1,2)]
```

```
y2 =

  Columns 1 through 14

    3.9380    5.3777    9.0230   13.1683   15.8740   15.8740
    13.1683    9.0230    5.3777    3.9380    5.3777    9.0230
    13.1683   15.8740


  Columns 15 through 19

    15.8740   13.1683    9.0230    5.3777    3.9380

>> plot(x,y2)
>> xlabel('\theta (degrees)');
>> ylabel('Q^{-}_{12} (GPa)');
```

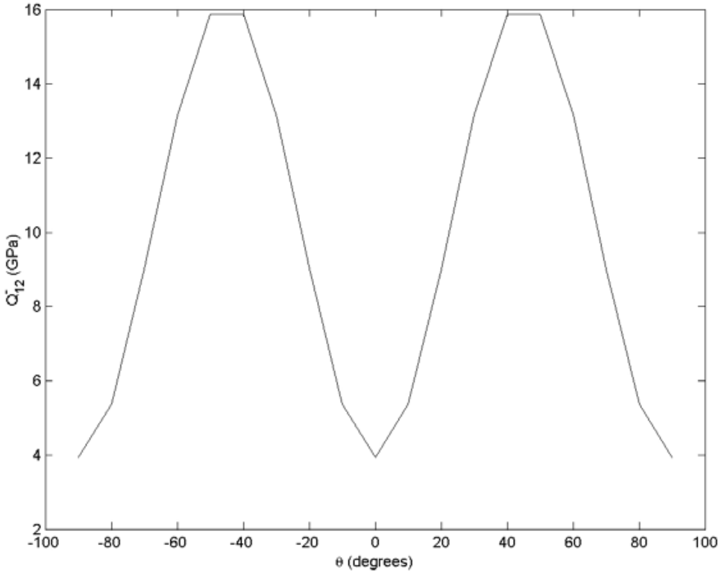

Fig. Variation of $\bar{Q}_{12}$ versus θ for Problem 5.10

```
>> y3 = [Q1(1,3) Q2(1,3) Q3(1,3) Q4(1,3) Q5(1,3) Q6(1,3) Q7(1,3)
        Q8(1,3) Q9(1,3) Q10(1,3) Q11(1,3) Q12(1,3) Q13(1,3) Q14(1,3)
        Q15(1,3) Q16(1,3) Q17(1,3) Q18(1,3) Q19(1,3)]
```

```
y3 =

  Columns 1 through 14

     0.0000    1.8406    0.7118   -4.7121  -13.2692  -21.6877
   -26.0285  -23.5283  -13.9810         0   13.9810   23.5283
    26.0285   21.6877

  Columns 15 through 19

    13.2692    4.7121   -0.7118   -1.8406   -0.0000

>> plot(x,y3)
>> xlabel('\theta (degrees)');
>> ylabel('Q^{-}_{16} (GPa)');
```

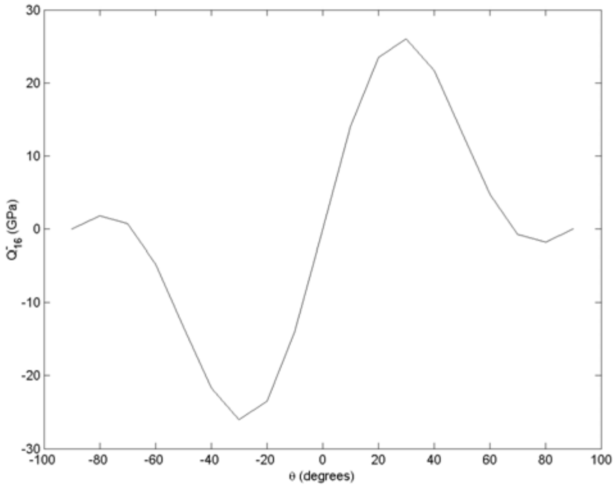

Fig. Variation of $\bar{Q}_{16}$ versus θ for Problem 5.10

```
>> y4 = [Q1(2,2) Q2(2,2) Q3(2,2) Q4(2,2) Q5(2,2) Q6(2,2) Q7(2,2)
        Q8(2,2) Q9(2,2) Q10(2,2) Q11(2,2) Q12(2,2) Q13(2,2) Q14(2,2)
        Q15(2,2) Q16(2,2) Q17(2,2) Q18(2,2) Q19(2,2)]

y4 =

  Columns 1 through 14

   51.0003   48.4903   41.7630   32.8959   24.3981   18.2343
   15.1478   14.5714   15.1348   15.5041   15.1348   14.5714
   15.1478   18.2343
```

```
  Columns 15 through 19

    24.3981   32.8959   41.7630   48.4903   51.0003

>> plot(x,y4)
>> xlabel('\theta (degrees)');
>> ylabel('Q^{-}_{22} (GPa)');
```

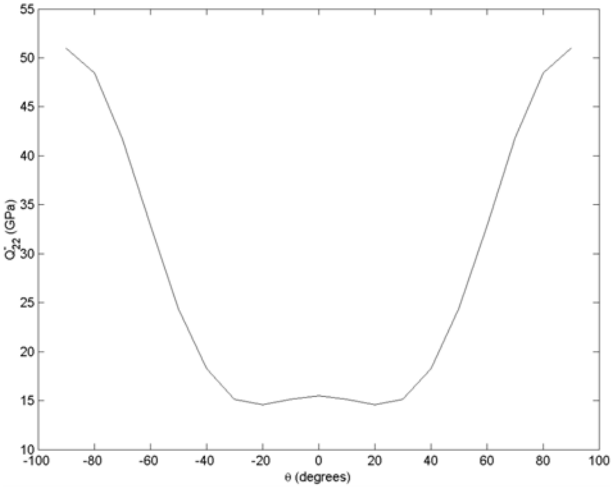

Fig. Variation of $\bar{Q}_{22}$ versus θ for Problem 5.10

```
>> y5 = [Q1(2,3) Q2(2,3) Q3(2,3) Q4(2,3) Q5(2,3) Q6(2,3) Q7(2,3)
        Q8(2,3) Q9(2,3) Q10(2,3) Q11(2,3) Q12(2,3) Q13(2,3) Q14(2,3)
        Q15(2,3) Q16(2,3) Q17(2,3) Q18(2,3) Q19(2,3)]

y5 =

  Columns 1 through 14

   -0.0000  -13.9810  -23.5283  -26.0285  -21.6877  -13.2692
   -4.7121    0.7118    1.8406         0   -1.8406   -0.7118
    4.7121   13.2692

  Columns 15 through 19

    21.6877   26.0285   23.5283   13.9810    0.0000

>> plot(x,y5)
>> xlabel('\theta (degrees)');
>> ylabel('Q^{-}_{26} (GPa)');
```

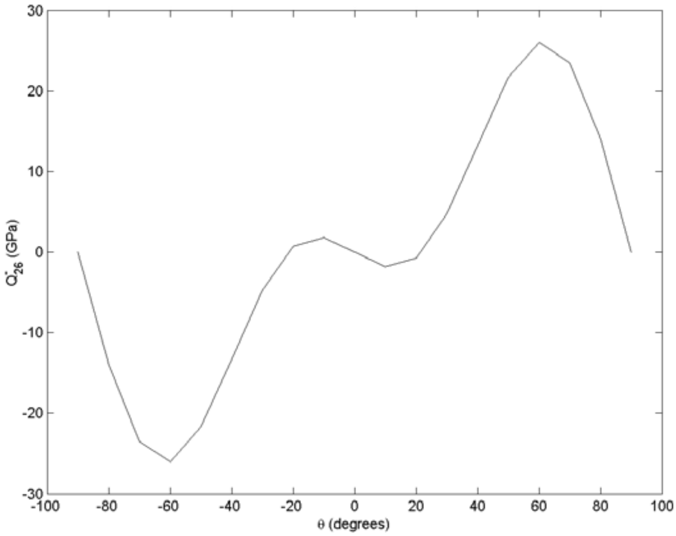

Fig. Variation of $\bar{Q}_{26}$ versus θ for Problem 5.10

```
>> y6 = [Q1(3,3) Q2(3,3) Q3(3,3) Q4(3,3) Q5(3,3) Q6(3,3) Q7(3,3)
        Q8(3,3) Q9(3,3) Q10(3,3) Q11(3,3) Q12(3,3) Q13(3,3) Q14(3,3)
        Q15(3,3) Q16(3,3) Q17(3,3) Q18(3,3) Q19(3,3)]

y6 =

  Columns 1 through 14

    4.7000    7.5793   14.8700   23.1606   28.5719   28.5719
   23.1606   14.8700    7.5793    4.7000    7.5793   14.8700
   23.1606   28.5719

  Columns 15 through 19

   28.5719   23.1606   14.8700    7.5793    4.7000

>> plot(x,y6)
>> xlabel('\theta (degrees)');
>> ylabel('Q^{-}_{66} (GPa)');
```

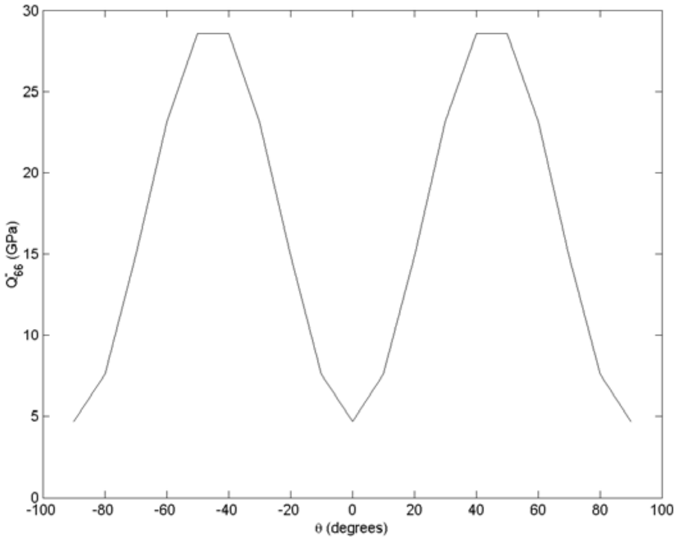

Fig. Variation of $\bar{Q}_{66}$ versus θ for Problem 5.10

Problem 5.11

When $\theta = 0°$, we have $[T] = [T]^{-1} = [I]$, where $[I]$ is the identity matrix. Therefore, we have;

$$[\bar{S}] = [T]^{-1}[S][T] = [I][S][I] = [S]$$
$$[\bar{Q}] = [T]^{-1}[Q][T] = [I][Q][I] = [Q]$$

Problem 5.12

For isotropic materials, we showed in Problem 4.3 that $[S]$ is given by:

$$[S] = \begin{bmatrix} \dfrac{1}{E} & \dfrac{-\nu}{E} & 0 \\ \dfrac{-\nu}{E} & \dfrac{1}{E} & 0 \\ 0 & 0 & \dfrac{2(1+\nu)}{E} \end{bmatrix}$$

Therefore, we have:

$$S_{11} = \frac{1}{E}$$
$$S_{12} = \frac{-\nu}{E}$$

$$
\begin{aligned}
S_{22} &= \frac{1}{E} \\
S_{16} &= 0 \\
S_{26} &= 0 \\
S_{66} &= \frac{2(1+\nu)}{E}
\end{aligned}
$$

Substitute the above equations into (5.16) from the book to obtain:

$$
\begin{aligned}
\bar{S}_{11} &= \frac{1}{E}m^4 + \left[\frac{-2\nu}{E} + \frac{2(1+\nu)}{E}\right] n^2m^2 + \frac{1}{E}n^4 \\
&= \frac{1}{E}\left(m^2+n^2\right)^2 \\
&= \frac{1}{E} \\
\bar{S}_{12} &= \left[\frac{1}{E} + \frac{1}{E} - \frac{2(1+\nu)}{E}\right] n^2m^2 - \frac{\nu}{E}\left(n^4+m^4\right) \\
&= -\frac{\nu}{E}\left(m^2+n^2\right)^2 \\
&= -\frac{\nu}{E} \\
\bar{S}_{22} &= \frac{1}{E} \text{ (derivation similar to } \bar{S}_{11}\text{).} \\
\bar{S}_{16} &= \left[\frac{2}{E} - \frac{2\nu}{E} - \frac{2(1+\nu)}{E}\right] nm^3 - \left[\frac{2}{E} - \frac{2\nu}{E} - \frac{2(1+\nu)}{E}\right] n^3m \\
&= 0 - 0 \\
&= 0 \\
\bar{S}_{26} &= 0 \text{ (derivation similar to } \bar{S}_{16}\text{).} \\
\bar{S}_{66} &= 2\left[\frac{2}{E} + \frac{2}{E} + \frac{4\nu}{E} - \frac{2(1+\nu)}{E}\right] n^2m^2 + \frac{2(1+\nu)}{E}\left(n^4+m^4\right) \\
&= \frac{2(1+\nu)}{E}\left(m^2+n^2\right)^2 \\
&= \frac{2(1+\nu)}{E}
\end{aligned}
$$

Therefore, we have now the following equation;

$$[\bar{S}] = [S] = \begin{bmatrix} \dfrac{1}{E} & \dfrac{-\nu}{E} & 0 \\ \dfrac{-\nu}{E} & \dfrac{1}{E} & 0 \\ 0 & 0 & \dfrac{2(1+\nu)}{E} \end{bmatrix}$$

Problem 5.13

We can follow the same approach used in solving Problem 5.12 while using the result of Problem 5.5. Alternatively, we can follow a shorter approach by using Problem 5.4 and taking the inverse of $[\bar{S}]$ as follows:

From Problem 5.12, we have:

$$[\bar{S}] = \begin{bmatrix} \dfrac{1}{E} & \dfrac{-\nu}{E} & 0 \\ \dfrac{-\nu}{E} & \dfrac{1}{E} & 0 \\ 0 & 0 & \dfrac{2(1+\nu)}{E} \end{bmatrix}$$

and from Problem 5.5 we obtain:

$$[\bar{Q}] = [\bar{S}]^{-1} = \begin{bmatrix} \dfrac{1}{E} & \dfrac{-\nu}{E} & 0 \\ \dfrac{-\nu}{E} & \dfrac{1}{E} & 0 \\ 0 & 0 & \dfrac{2(1+\nu)}{E} \end{bmatrix}^{-1} = \begin{bmatrix} \dfrac{E}{1-\nu^2} & \dfrac{\nu E}{1-\nu^2} & 0 \\ \dfrac{\nu E}{1-\nu^2} & \dfrac{E}{1-\nu^2} & 0 \\ 0 & 0 & \dfrac{E}{2(1+\nu)} \end{bmatrix} = [Q]$$

See also Problem 4.4.

Problem 5.14

```
>> S = ReducedCompliance(50.0, 15.20, 0.254, 4.70)

S =

    0.0200   -0.0051         0
   -0.0051    0.0658         0
         0         0    0.2128

>> S1 = Sbar(S,0)
```

```
S1 =

    0.0200   -0.0051         0
   -0.0051    0.0658         0
         0         0    0.2128

>> sigma = [100e-3 ; 0 ; 0]

sigma =

    0.1000
         0
         0

>> epsilon = S1*sigma

epsilon =

    0.0020
   -0.0005
         0

>> deltax = 50*epsilon(1)

deltax =

    0.1000

>> deltay = 50*epsilon(2)

deltay =

   -0.0254

>> gammaxy = epsilon(3)

gammaxy =

     0

>> dx = 50 + deltax

dx =

   50.1000
```

```
>> dy = 50 + deltay

dy =

   49.9746

>> S2 = Sbar(S, 45)

S2 =

    0.1253   -0.0875   -0.0229
   -0.0875    0.1253   -0.0229
   -0.0114   -0.0114    0.0480

>> epsilon = S2*sigma

epsilon =

    0.0125
   -0.0087
   -0.0011

>> deltax = 50*epsilon(1)

deltax =

    0.6265

>> deltay = 50*epsilon(2)

deltay =

   -0.4374

>> dx = 50 + deltax

dx =

   50.6265

>> dy = 50 + deltay

dy =

   49.5626

>> gammaxy = epsilon(3)
```

```
gammaxy =

   -0.0011

>> S3 = Sbar(S, -45)

S3 =

    0.1253   -0.0875    0.0229
   -0.0875    0.1253    0.0229
    0.0114    0.0114    0.0480

>> epsilon = S3*sigma

epsilon =

    0.0125
   -0.0087
    0.0011

>> deltax = 50*epsilon(1)

deltax =

    0.6265

>> deltay = 50*epsilon(2)

deltay =

   -0.4374

>> dy = 50 + deltay

dy =

   49.5626

>> dx = 50 + deltax

dx =

   50.6265

>> gammaxy = epsilon(3)

gammaxy =

    0.0011
```

Problem 5.15

Using the result of Problem 4.10, we have:

$$\begin{Bmatrix} \varepsilon_1 - \alpha_1 \Delta T - \beta_1 \Delta M \\ \varepsilon_2 - \alpha_2 \Delta T - \beta_2 \Delta M \\ \gamma_{12} \end{Bmatrix} = \begin{bmatrix} S_{11} & S_{12} & 0 \\ S_{12} & S_{22} & 0 \\ 0 & 0 & S_{66} \end{bmatrix} \begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{Bmatrix}$$

Now, we need to transform the above equation from the 1-2-3 coordinate system to the x-y-z global coordinate system. The above equation can be re-written as follows where we have introduced a factor of 1/2 for the engineering shear strain:

$$\begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \frac{1}{2}\gamma_{12} \end{Bmatrix} - \begin{Bmatrix} \alpha_1 \Delta T \\ \alpha_2 \Delta T \\ \frac{0}{2} \end{Bmatrix} - \begin{Bmatrix} \beta_1 \Delta M \\ \beta_2 \Delta M \\ \frac{0}{2} \end{Bmatrix} = \begin{bmatrix} S_{11} & S_{12} & 0 \\ S_{12} & S_{22} & 0 \\ 0 & 0 & S_{66} \end{bmatrix} \begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{Bmatrix}$$

Next, we substitute the following transformation relations along with (5.2) and (5.6) into the above equation:

$$\begin{Bmatrix} \alpha_1 \Delta T \\ \alpha_2 \Delta T \\ \frac{0}{2} \end{Bmatrix} = [T] \begin{Bmatrix} \alpha_x \Delta T \\ \alpha_y \Delta T \\ \frac{1}{2}\alpha_{xy} \Delta T \end{Bmatrix}$$

$$\begin{Bmatrix} \beta_1 \Delta M \\ \beta_2 \Delta M \\ \frac{0}{2} \end{Bmatrix} = [T] \begin{Bmatrix} \beta_x \Delta M \\ \beta_y \Delta M \\ \frac{1}{2}\beta_{xy} \Delta M \end{Bmatrix}$$

Therefore, we obtain the desired relation as follows (after grouping the terms together and using (5.11)):

$$\begin{Bmatrix} \varepsilon_x - \alpha_x \Delta T - \beta_x \Delta M \\ \varepsilon_y - \alpha_y \Delta T - \beta_y \Delta M \\ \gamma_{xy} - \alpha_{xy} \Delta T - \beta_{xy} \Delta M \end{Bmatrix} = \begin{bmatrix} \bar{S}_{11} & \bar{S}_{12} & \bar{S}_{16} \\ \bar{S}_{12} & \bar{S}_{22} & \bar{S}_{26} \\ \bar{S}_{16} & \bar{S}_{26} & \bar{S}_{66} \end{bmatrix} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix}$$

Taking the inverse of the above relation, we obtain the second desired results as follows:

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \begin{bmatrix} \bar{Q}_{11} & \bar{Q}_{12} & \bar{Q}_{16} \\ \bar{Q}_{12} & \bar{Q}_{22} & \bar{Q}_{26} \\ \bar{Q}_{16} & \bar{Q}_{26} & \bar{Q}_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_x - \alpha_x \Delta T - \beta_x \Delta M \\ \varepsilon_y - \alpha_y \Delta T - \beta_y \Delta M \\ \gamma_{xy} - \alpha_{xy} \Delta T - \beta_{xy} \Delta M \end{Bmatrix}$$

Problem 6.1

From an elementary course on mechanics of materials, we have the following equation:

$$\nu_{xy} = -\frac{\varepsilon_y}{\varepsilon_x}$$

We also have the following two equations that can be obtained from (5.10):

$$\varepsilon_y = \bar{S}_{12}\sigma_x$$
$$\varepsilon_x = \bar{S}_{11}\sigma_x$$

Substitute the above two equations into the first equation above to obtain the desired relation:

$$\nu_{xy} = -\frac{\bar{S}_{12}}{\bar{S}_{11}} = \frac{\nu_{12}\left(n^4 + m^4\right) - \left(1 + \frac{E_1}{E_2} - \frac{E_1}{G_{12}}\right) n^2 m^2}{m^4 + \left(\frac{E_1}{G_{12}} - 2\nu_{12}\right) n^2 m^2 + \frac{E_1}{E_2} n^2}$$

where we have used (5.16) from Chap. 5.

Problem 6.2

From an elementary course on mechanics of materials, we have the following equation:

$$\varepsilon_y = \frac{\sigma_y}{E_y}$$

We also have the following equation that can be obtained from (5.10):

$$\varepsilon_y = \bar{S}_{22}\sigma_y$$

Comparing the above two equation, we obtain the desired result as follows:

$$E_y = \frac{1}{\bar{S}_{22}} = \frac{E_2}{m^4 + \left(\frac{E_2}{G_{12}} - 2\nu_{21}\right) n^2 m^2 + \frac{E_2}{E_1} n^4}$$

where we have used (5.16) from Chap. 5.

Problem 6.3

From an elementary course on mechanics of materials, we have the following equation:

$$\nu_{yx} = -\frac{\varepsilon_x}{\varepsilon_y}$$

We also have the following two equations that can be obtained from (5.10):

$$\varepsilon_y = \bar{S}_{22}\sigma_y$$
$$\varepsilon_x = \bar{S}_{12}\sigma_y$$

Substitute the above two equations into the first equation above to obtain the desired relation:

$$\nu_{yx} = -\frac{\bar{S}_{12}}{\bar{S}_{22}} = \frac{\nu_{21}\left(n^4 + m^4\right) - \left(1 + \frac{E_2}{E_1} - \frac{E_2}{G_{12}}\right) n^2 m^2}{m^4 + \left(\frac{E_2}{G_{12}} - 2\nu_{21}\right) n^2 m^2 + \frac{E_2}{E_1} n^2}$$

where we have used (5.16) from Chap. 5.

Problem 6.4

From an elementary course on mechanics of materials, we have the following equation:

$$\gamma_{xy} = \frac{\tau_{xy}}{G_{xy}}$$

We also have the following equation which can be obtained from (5.10):

$$\gamma_{xy} = \bar{S}_{66} \tau_{xy}$$

Comparing the above two equation, we obtain the desired result as follows:

$$G_{xy} = \frac{1}{\bar{S}_{66}} = \frac{G_{12}}{n^4 + m^4 + 2\left(\frac{2G_{12}}{E_1}\left(1 + 2\nu_{12}\right) + \frac{2G_{12}}{E_2} - 1\right) n^2 m^2}$$

where we have used (5.16) from Chap. 5.

Problem 6.5

```
>> Ex1 = Ex(50.0, 15.20, 0.254, 4.70, -90)

Ex1 =

   15.2000

>> Ex2 = Ex(50.0, 15.20, 0.254, 4.70, -80)

Ex2 =

   14.7438
```

```
>> Ex3 = Ex(50.0, 15.20, 0.254, 4.70, -70)

Ex3 =

   13.7932

>> Ex4 = Ex(50.0, 15.20, 0.254, 4.70, -60)

Ex4 =

   13.1156

>> Ex5 = Ex(50.0, 15.20, 0.254, 4.70, -50)

Ex5 =

   13.2990

>> Ex6 = Ex(50.0, 15.20, 0.254, 4.70, -40)

Ex6 =

   14.8715

>> Ex7 = Ex(50.0, 15.20, 0.254, 4.70, -30)

Ex7 =

   18.7440

>> Ex8 = Ex(50.0, 15.20, 0.254, 4.70, -20)

Ex8 =

   26.7217

>> Ex9 = Ex(50.0, 15.20, 0.254, 4.70, -10)

Ex9 =

   40.3275
```

```
>> Ex10 = Ex(50.0, 15.20, 0.254, 4.70, 0)

Ex10 =

    50

>> Ex11 = Ex(50.0, 15.20, 0.254, 4.70, 10)

Ex11 =

   40.3275

>> Ex12 = Ex(50.0, 15.20, 0.254, 4.70, 20)

Ex12 =

   26.7217

>> Ex13 = Ex(50.0, 15.20, 0.254, 4.70, 30)

Ex13 =

   18.7440

>> Ex14 = Ex(50.0, 15.20, 0.254, 4.70, 40)

Ex14 =

   14.8715

>> Ex15 = Ex(50.0, 15.20, 0.254, 4.70, 50)

Ex15 =

   13.2990

>> Ex16 = Ex(50.0, 15.20, 0.254, 4.70, 60)

Ex16 =

   13.1156

>> Ex17 = Ex(50.0, 15.20, 0.254, 4.70, 70)

Ex17 =

   13.7932
```

```
>> Ex18 = Ex(50.0, 15.20, 0.254, 4.70, 80)

Ex18 =

   14.7438


>> Ex19 = Ex(50.0, 15.20, 0.254, 4.70, 90)

Ex19 =

   15.2000

>> y1 = [Ex1 Ex2 Ex3 Ex4 Ex5 Ex6 Ex7 Ex8 Ex9 Ex10 Ex11 Ex12 Ex13 Ex14
        Ex15 Ex16 Ex17 Ex18 Ex19]

y1 =

  Columns 1 through 14

    15.2000   14.7438   13.7932   13.1156   13.2990   14.8715
    18.7440   26.7217   40.3275   50.0000   40.3275   26.7217
    18.7440   14.8715

  Columns 15 through 19

    13.2990   13.1156   13.7932   14.7438   15.2000

>> x = [-90 -80 -70 -60 -50 -40 -30 -20 -10 0 10 20 30 40 50 60 70
       80 90]

x =

   -90   -80   -70   -60   -50   -40   -30   -20   -10     0    10
    20    30    40    50    60    70    80    90

>> plot(x,y1)
>> xlabel('\theta (degrees)');
>> ylabel('E_x (GPa)');
```

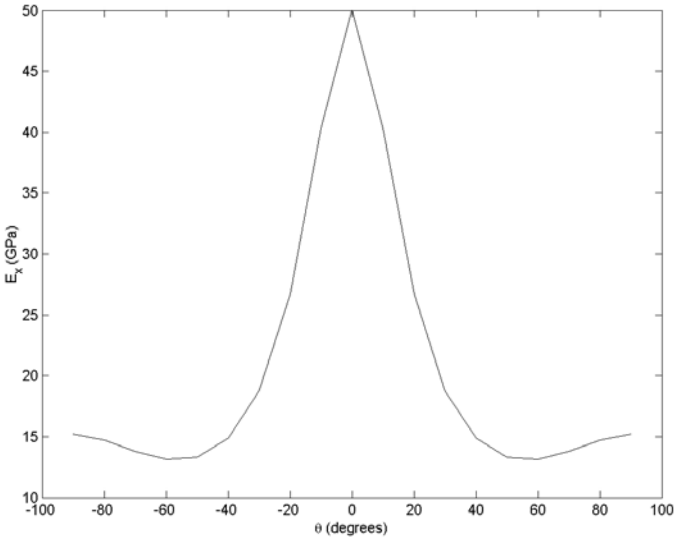

Fig. Variation of E_x versus θ for Problem 6.5

```
>> NUxy1 = NUxy(50.0, 15.20, 0.254, 4.70, -90)

NUxy1 =

    0.0772

>> NUxy2 = NUxy(50.0, 15.20, 0.254, 4.70, -80)

NUxy2 =

    0.1218

>> NUxy3 = NUxy(50.0, 15.20, 0.254, 4.70, -70)


NUxy3 =

    0.2162

>> NUxy4 = NUxy(50.0, 15.20, 0.254, 4.70, -60)

NUxy4 =

    0.3046
```

```
>> NUxy5 = NUxy(50.0, 15.20, 0.254, 4.70, -50)

NUxy5 =

    0.3665

>> NUxy6 = NUxy(50.0, 15.20, 0.254, 4.70, -40)

NUxy6 =

    0.4015

>> NUxy7 = NUxy(50.0, 15.20, 0.254, 4.70, -30)

NUxy7 =

    0.4108
>> NUxy8 = NUxy(50.0, 15.20, 0.254, 4.70, -20)

NUxy8 =

    0.3878

>> NUxy9 = NUxy(50.0, 15.20, 0.254, 4.70, -10)

NUxy9 =

    0.3180

>> NUxy10 = NUxy(50.0, 15.20, 0.254, 4.70, 0)

NUxy10 =

    0.2540

>> NUxy11 = NUxy(50.0, 15.20, 0.254, 4.70, 10)

NUxy11 =

    0.3180

>> NUxy12 = NUxy(50.0, 15.20, 0.254, 4.70, 20)

NUxy12 =

    0.3878
```

```
>> NUxy13 = NUxy(50.0, 15.20, 0.254, 4.70, 30)

NUxy13 =

    0.4108

>> NUxy14 = NUxy(50.0, 15.20, 0.254, 4.70, 40)

NUxy14 =

    0.4015

>> NUxy15 = NUxy(50.0, 15.20, 0.254, 4.70, 50)

NUxy15 =

    0.3665


>> NUxy16 = NUxy(50.0, 15.20, 0.254, 4.70, 60)

NUxy16 =

    0.3046

>> NUxy17 = NUxy(50.0, 15.20, 0.254, 4.70, 70)

NUxy17 =

    0.2162

>> NUxy18 = NUxy(50.0, 15.20, 0.254, 4.70, 80)

NUxy18 =

    0.1218

>> NUxy19 = NUxy(50.0, 15.20, 0.254, 4.70, 90)

NUxy19 =

    0.0772

>> y2 = [NUxy1 NUxy2 NUxy3 NUxy4 NUxy5 NUxy6 NUxy7 NUxy8 NUxy9 NUxy10
        NUxy11 NUxy12 NUxy13 NUxy14 NUxy15 NUxy16 NUxy17 NUxy18 NUxy19]
```

```
y2 =

  Columns 1 through 14

    0.0772    0.1218    0.2162    0.3046    0.3665    0.4015
    0.4108    0.3878    0.3180    0.2540    0.3180    0.3878
    0.4108    0.4015

  Columns 15 through 19

    0.3665    0.3046    0.2162    0.1218    0.0772

>> plot(x,y2)
>> xlabel('\theta (degrees)');
>> ylabel('\nu_{xy}');
```

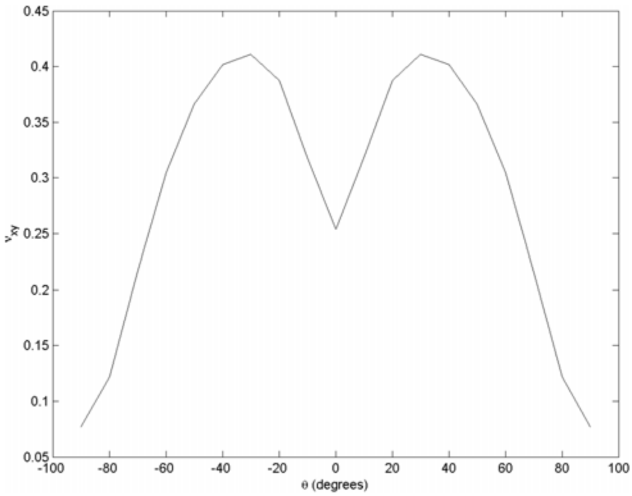

Fig. Variation of ν_{xy} versus θ for Problem 6.5

```
>> Ey1 = Ey(50.0, 15.20, 0.254, 4.70, -90)

Ey1 =

    50

>> Ey2 = Ey(50.0, 15.20, 0.254, 4.70, -80)

Ey2 =

   41.4650
```

```
>> Ey3 = Ey(50.0, 15.20, 0.254, 4.70, -70)

Ey3 =

   28.5551

>> Ey4 = Ey(50.0, 15.20, 0.254, 4.70, -60)

Ey4 =

   20.4127

>> Ey5 = Ey(50.0, 15.20, 0.254, 4.70, -50)

Ey5 =

   16.2331

>> Ey6 = Ey(50.0, 15.20, 0.254, 4.70, -40)

Ey6 =

   14.3773

>> Ey7 = Ey(50.0, 15.20, 0.254, 4.70, -30)

Ey7 =

   13.9114
>> Ey8 = Ey(50.0, 15.20, 0.254, 4.70, -20)

Ey8 =

   14.2660

>> Ey9 = Ey(50.0, 15.20, 0.254, 4.70, -10)

Ey9 =

   14.8932

>> Ey10 = Ey(50.0, 15.20, 0.254, 4.70, 0)

Ey10 =

   15.2000
```

```
>> Ey11 = Ey(50.0, 15.20, 0.254, 4.70, 10)

Ey11 =

   14.8932

>> Ey12 = Ey(50.0, 15.20, 0.254, 4.70, 20)

Ey12 =

   14.2660

>> Ey13 = Ey(50.0, 15.20, 0.254, 4.70, 30)

Ey13 =

   13.9114

>> Ey14 = Ey(50.0, 15.20, 0.254, 4.70, 40)

Ey14 =

   14.3773

>> Ey15 = Ey(50.0, 15.20, 0.254, 4.70, 50)

Ey15 =

   16.2331

>> Ey16 = Ey(50.0, 15.20, 0.254, 4.70, 60)

Ey16 =

   20.4127

>> Ey17 = Ey(50.0, 15.20, 0.254, 4.70, 70)

Ey17 =

   28.5551

>> Ey18 = Ey(50.0, 15.20, 0.254, 4.70, 80)

Ey18 =

   41.4650
```

```
>> Ey19 = Ey(50.0, 15.20, 0.254, 4.70, 90)

Ey19 =

    50

>> y3 = [Ey1 Ey2 Ey3 Ey4 Ey5 Ey6 Ey7 Ey8 Ey9 Ey10 Ey11 Ey12 Ey13 Ey14
        Ey15 Ey16 Ey17 Ey18 Ey19]

y3 =

  Columns 1 through 14

    50.0000   41.4650   28.5551   20.4127   16.2331   14.3773
    13.9114   14.2660   14.8932   15.2000   14.8932   14.2660
    13.9114   14.3773

  Columns 15 through 19

    16.2331   20.4127   28.5551   41.4650   50.0000

>> plot(x,y3)
>> xlabel('\theta (degrees)');
>> ylabel('E_y (GPa)');
```

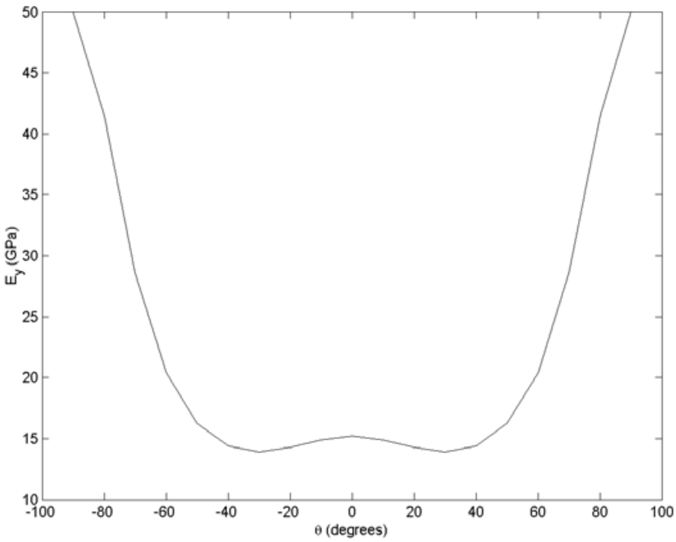

Fig. Variation of E_y versus θ for Problem 6.5

```
>> NUyx1 = NUyx(50.0, 15.20, 0.254, 4.70, -90)

NUyx1 =

    0.8355

>> NUyx2 = NUyx(50.0, 15.20, 0.254, 4.70, -80)

NUyx2 =

    0.7873

>> NUyx3 = NUyx(50.0, 15.20, 0.254, 4.70, -70)

NUyx3 =

    0.7112

>> NUyx4 = NUyx(50.0, 15.20, 0.254, 4.70, -60)

NUyx4 =

    0.6495

>> NUyx5 = NUyx(50.0, 15.20, 0.254, 4.70, -50)

NUyx5 =

    0.5928

>> NUyx6 = NUyx(50.0, 15.20, 0.254, 4.70, -40)

NUyx6 =

    0.5295

>> NUyx7 = NUyx(50.0, 15.20, 0.254, 4.70, -30)

NUyx7 =

    0.4529

>> NUyx8 = NUyx(50.0, 15.20, 0.254, 4.70, -20)

NUyx8 =

    0.3655
```

```
>> NUyx9 = NUyx(50.0, 15.20, 0.254, 4.70, -10)

NUyx9 =

    0.2871

>> NUyx10 = NUyx(50.0, 15.20, 0.254, 4.70, 0)

NUyx10 =

    0.2540

>> NUyx11 = NUyx(50.0, 15.20, 0.254, 4.70, 10)

NUyx11 =

    0.2871

>> NUyx12 = NUyx(50.0, 15.20, 0.254, 4.70, 20)

NUyx12 =

    0.3655

>> NUyx13 = NUyx(50.0, 15.20, 0.254, 4.70, 30)

NUyx13 =

    0.4529

>> NUyx14 = NUyx(50.0, 15.20, 0.254, 4.70, 40)

NUyx14 =

    0.5295

>> NUyx15 = NUyx(50.0, 15.20, 0.254, 4.70, 50)

NUyx15 =

    0.5928

>> NUyx16 = NUyx(50.0, 15.20, 0.254, 4.70, 60)

NUyx16 =

    0.6495
```

```
>> NUyx17 = NUyx(50.0, 15.20, 0.254, 4.70, 70)

NUyx17 =

    0.7112

>> NUyx18 = NUyx(50.0, 15.20, 0.254, 4.70, 80)

NUyx18 =

    0.7873

>> NUyx19 = NUyx(50.0, 15.20, 0.254, 4.70, 90)

NUyx19 =

    0.8355

>> y4 = [NUyx1 NUyx2 NUyx3 NUyx4 NUyx5 NUyx6 NUyx7 NUyx8 NUyx9 NUyx10
        NUyx11 NUyx12 NUyx13 NUyx14 NUyx15 NUyx16 NUyx17 NUyx18 NUyx19]


y4 =

 Columns 1 through 14

    0.8355    0.7873    0.7112    0.6495    0.5928    0.5295
    0.4529    0.3655    0.2871    0.2540    0.2871    0.3655
    0.4529    0.5295

  Columns 15 through 19

    0.5928    0.6495    0.7112    0.7873    0.8355

>> plot(x,y4)
>> xlabel('\theta (degrees)');
>> ylabel('\nu_{yx}');
```

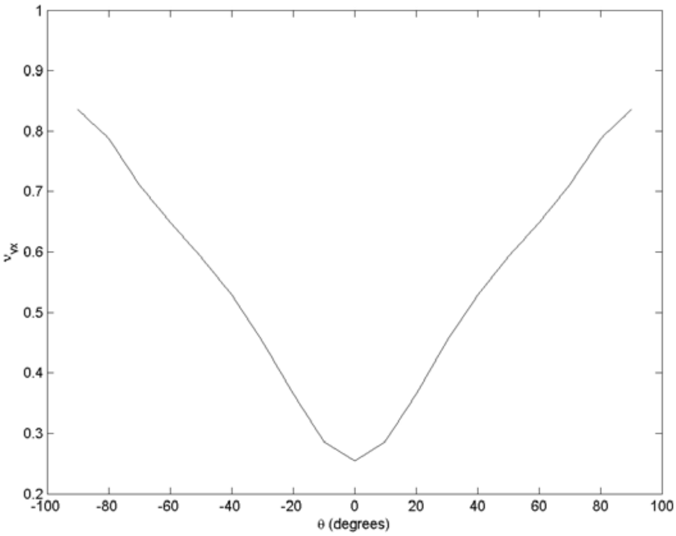

Fig. Variation of ν_{yx} versus θ for Problem 6.5

```
>> Gxy1 = Gxy(50.0, 15.20, 0.254, 4.70, -90)

Gxy1 =

    4.7000

>> Gxy2 = Gxy(50.0, 15.20, 0.254, 4.70, -80)

Gxy2 =

    5.0226

>> Gxy3 = Gxy(50.0, 15.20, 0.254, 4.70, -70)

Gxy3 =

    6.0790

>> Gxy4 = Gxy(50.0, 15.20, 0.254, 4.70, -60)

Gxy4 =

    7.9902
```

```
>> Gxy5 = Gxy(50.0, 15.20, 0.254, 4.70, -50)

Gxy5 =

   10.0531

>> Gxy6 = Gxy(50.0, 15.20, 0.254, 4.70, -40)

Gxy6 =

   10.0531

>> Gxy7 = Gxy(50.0, 15.20, 0.254, 4.70, -30)

Gxy7 =

    7.9902

>> Gxy8 = Gxy(50.0, 15.20, 0.254, 4.70, -20)

Gxy8 =

    6.0790

>> Gxy9 = Gxy(50.0, 15.20, 0.254, 4.70, -10)

Gxy9 =

    5.0226

>> Gxy10 = Gxy(50.0, 15.20, 0.254, 4.70, 0)

Gxy10 =

    4.7000

>> Gxy11 = Gxy(50.0, 15.20, 0.254, 4.70, 10)

Gxy11 =

    5.0226

>> Gxy12 = Gxy(50.0, 15.20, 0.254, 4.70, 20)

Gxy12 =

    6.0790
```

```
>> Gxy13 = Gxy(50.0, 15.20, 0.254, 4.70, 30)

Gxy13 =

    7.9902

>> Gxy14 = Gxy(50.0, 15.20, 0.254, 4.70, 40)

Gxy14 =

   10.0531

>> Gxy15 = Gxy(50.0, 15.20, 0.254, 4.70, 50)

Gxy15 =

   10.0531

>> Gxy16 = Gxy(50.0, 15.20, 0.254, 4.70, 60)

Gxy16 =

    7.9902

>> Gxy17 = Gxy(50.0, 15.20, 0.254, 4.70, 70)

Gxy17 =

    6.0790

>> Gxy18 = Gxy(50.0, 15.20, 0.254, 4.70, 80)

Gxy18 =

    5.0226

>> Gxy19 = Gxy(50.0, 15.20, 0.254, 4.70, 90)

Gxy19 =

    4.7000

>> y5 = [Gxy1 Gxy2 Gxy3 Gxy4 Gxy5 Gxy6 Gxy7 Gxy8 Gxy9 Gxy10 Gxy11
        Gxy12 Gxy13 Gxy14 Gxy15 Gxy16 Gxy17 Gxy18 Gxy19]
```

```
y5 =

  Columns 1 through 14

    4.7000    5.0226    6.0790    7.9902   10.0531   10.0531
    7.9902    6.0790    5.0226    4.7000    5.0226    6.0790
    7.9902   10.0531

  Columns 15 through 19

    10.0531    7.9902    6.0790    5.0226    4.7000

>> plot(x,y5)
>> xlabel('\theta (degrees)');
>> ylabel('G_{xy} (GPa)');
```

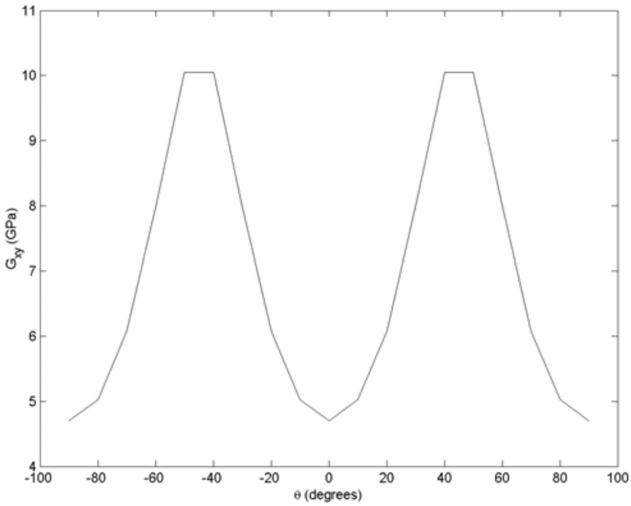

Fig. Variation of G_{xy} versus θ for Problem 6.5

Problem 6.6

From (5.10), we have:

$$\gamma_{xy} = \bar{S}_{16}\sigma_x$$
$$\varepsilon_x = \bar{S}_{11}\sigma_x$$

Substitute the above two equations into (6.6) to obtain the desired result as follows:

$$\eta_{xy,x} = \frac{\bar{S}_{16}}{\bar{S}_{11}}$$

Similarly, from (5.10) again, we have:

$$\gamma_{xy} = \bar{S}_{26}\sigma_y$$
$$\varepsilon_y = \bar{S}_{22}\sigma_y$$

Substitute the above two equation into (6.7) to obtain the desired result as follows:

$$\eta_{xy,y} = \frac{\bar{S}_{26}}{\bar{S}_{22}}$$

Problem 6.7

From (5.10), we have:

$$\varepsilon_x = \bar{S}_{16}\tau_{xy}$$
$$\gamma_{xy} = \bar{S}_{66}\tau_{xy}$$

Substitute the above two equations into (6.10) to obtain the desired result as follows:

$$\eta_{x,xy} = \frac{\bar{S}_{16}}{\bar{S}_{66}}$$

Similarly, from (5.10) again, we have:

$$\varepsilon_y = \bar{S}_{26}\tau_{xy}$$
$$\gamma_{xy} = \bar{S}_{66}\tau_{xy}$$

Substitute the above two equations into (6.11) to obtain the desired result as follows:

$$\eta_{y,xy} = \frac{\bar{S}_{26}}{\bar{S}_{66}}$$

Problem 6.8

Continuing with the commands from Example 6.3, we obtain:

```
>> Etaxxy1 = Etaxxy(S1)

Etaxxy1 =

  -7.7070e-017
```

```
>> Etaxxy2 = Etaxxy(S2)

Etaxxy2 =

   -0.2192

>> Etaxxy3 = Etaxxy(S3)

Etaxxy3 =

   -0.4244

>> Etaxxy4 = Etaxxy(S4)

Etaxxy4 =

   -0.4970

>> Etaxxy5 = Etaxxy(S5)

Etaxxy5 =

    0.1268

>> Etaxxy6 = Etaxxy(S6)

Etaxxy6 =

    1.3271

>> Etaxxy7 = Etaxxy(S7)

Etaxxy7 =

    1.2187

>> Etaxxy8 = Etaxxy(S8)

Etaxxy8 =

    0.7457

>> Etaxxy9 = Etaxxy(S9)

Etaxxy9 =

    0.3457
```

```
>> Etaxxy10 = Etaxxy(S10)

Etaxxy10 =

     0

>> Etaxxy11 = Etaxxy(S11)

Etaxxy11 =

   -0.3457

>> Etaxxy12 = Etaxxy(S12)

Etaxxy12 =

   -0.7457

>> Etaxxy13 = Etaxxy(S13)

Etaxxy13 =

   -1.2187

>> Etaxxy14 = Etaxxy(S14)

Etaxxy14 =

   -1.3271

>> Etaxxy15 = Etaxxy(S15)

Etaxxy15 =

   -0.1268

>> Etaxxy16 = Etaxxy(S16)

Etaxxy16 =

    0.4970

>> Etaxxy17 = Etaxxy(S17)

Etaxxy17 =

    0.4244
```

```
>> Etaxxy18 = Etaxxy(S18)

Etaxxy18 =

    0.2192

>> Etaxxy19 = Etaxxy(S19)

Etaxxy19 =

  7.7070e-017

>> y8 = [Etaxxy1 Etaxxy2 Etaxxy3 Etaxxy4 Etaxxy5 Etaxxy6 Etaxxy7
        Etaxxy8 Etaxxy9 Etaxxy10 Etaxxy11 Etaxxy12 Etaxxy13 Etaxxy14
        Etaxxy15 Etaxxy16 Etaxxy17 Etaxxy18 Etaxxy19]

y8 =

  Columns 1 through 14

    -0.0000   -0.2192   -0.4244   -0.4970    0.1268    1.3271
     1.2187    0.7457    0.3457         0   -0.3457   -0.7457   -1.2187
    -1.3271

  Columns 15 through 19

    -0.1268    0.4970    0.4244    0.2192    0.0000

>> plot(x,y8)
>> xlabel('\theta (degrees)');
>> ylabel('\eta_{x,xy}');

>> Etayxy1 = Etayxy(S1)

Etayxy1 =

  1.1813e-016

>> Etayxy2 = Etayxy(S2)

Etayxy2 =

    0.3457

>> Etayxy3 = Etayxy(S3)

Etayxy3 =

    0.7457
```

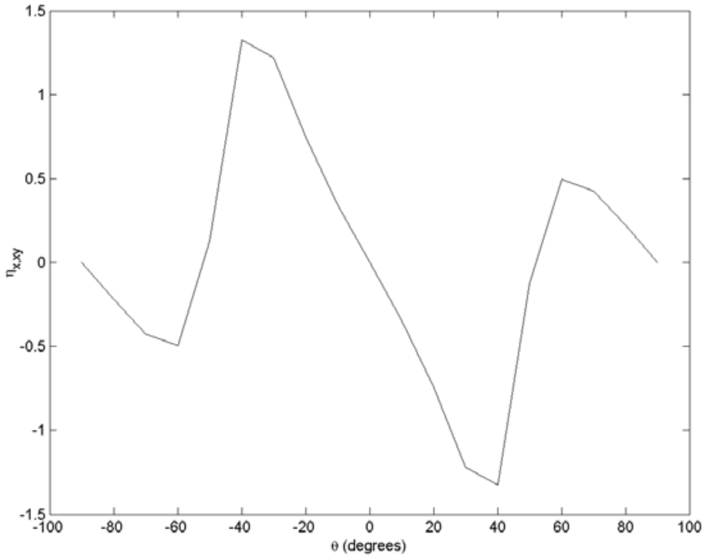

Fig. Variation of $\eta_{x,xy}$ versus θ for Problem 6.8

```
>> Etayxy4 = Etayxy(S4)

Etayxy4 =

    1.2187

>> Etayxy5 = Etayxy(S5)

Etayxy5 =

    1.3271

>> Etayxy6 = Etayxy(S6)

Etayxy6 =

    0.1268

>> Etayxy7 = Etayxy(S7)

Etayxy7 =

   -0.4970
```

```
>> Etayxy8 = Etayxy(S8)

Etayxy8 =

   -0.4244

>> Etayxy9 = Etayxy(S9)

Etayxy9 =

   -0.2192

>> Etayxy10 = Etayxy(S10)

Etayxy10 =

     0

>> Etayxy11 = Etayxy(S11)

Etayxy11 =

    0.2192

>> Etayxy12 = Etayxy(S12)

Etayxy12 =

    0.4244

>> Etayxy13 = Etayxy(S13)

Etayxy13 =

    0.4970

>> Etayxy14 = Etayxy(S14)

Etayxy14 =

   -0.1268

>> Etayxy15 = Etayxy(S15)

Etayxy15 =

   -1.3271
```

```
>> Etayxy16 = Etayxy(S16)

Etayxy16 =

   -1.2187

>> Etayxy17 = Etayxy(S17)

Etayxy17 =

   -0.7457

>> Etayxy18 = Etayxy(S18)

Etayxy18 =

   -0.3457

>> Etayxy19 = Etayxy(S19)

Etayxy19 =

 -1.1813e-016

>> y9 = [Etayxy1 Etayxy2 Etayxy3 Etayxy4 Etayxy5 Etayxy6 Etayxy7
        Etayxy8 Etayxy9 Etayxy10 Etayxy11 Etayxy12 Etayxy13 Etayxy14
        Etayxy15 Etayxy16 Etayxy17 Etayxy18 Etayxy19]

y9 =

  Columns 1 through 14

     0.0000    0.3457    0.7457    1.2187    1.3271    0.1268
    -0.4970   -0.4244   -0.2192         0    0.2192    0.4244
     0.4970   -0.1268

  Columns 15 through 19

    -1.3271   -1.2187   -0.7457   -0.3457   -0.0000

>> plot(x,y9)
>> xlabel('\theta {degrees}');
>> ylabel('\eta_{y,xy}');
```

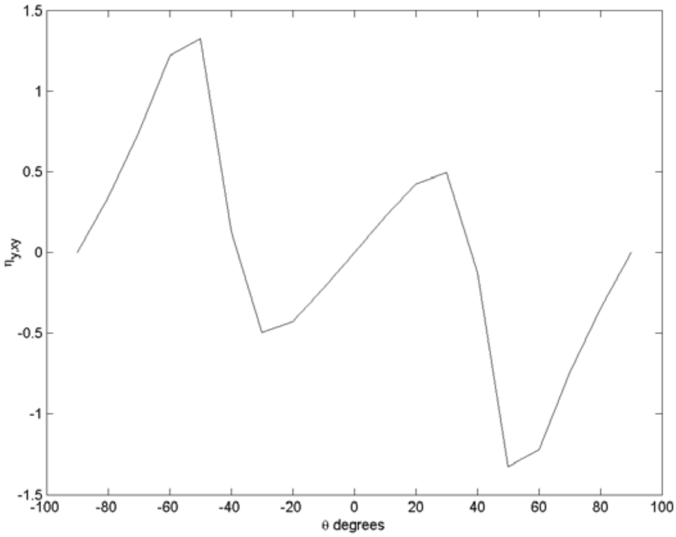

Fig. Variation of $\eta_{y,xy}$ versus θ for Problem 6.8

Problem 6.9

```
>> S = ReducedCompliance(50.0, 15.20, 0.254, 4.70)

S =

    0.0200   -0.0051         0
   -0.0051    0.0658         0
         0         0    0.2128

>> S1 = Sbar(S, -90)

S1 =

    0.0658   -0.0051   -0.0000
   -0.0051    0.0200    0.0000
   -0.0000    0.0000    0.2128
>> S2 = Sbar(S, -80)

S2 =

    0.0740   -0.0147   -0.0451
   -0.0147    0.0310    0.0608
   -0.0226    0.0304    0.1935
```

```
>> S3 = Sbar(S, -70)

S3 =

    0.0945   -0.0391   -0.0664
   -0.0391    0.0594    0.0959
   -0.0332    0.0479    0.1447

>> S4 = Sbar(S, -60)

S4 =

    0.1161   -0.0669   -0.0515
   -0.0669    0.0932    0.0912
   -0.0258    0.0456    0.0892

>> S5 = Sbar(S, -50)

S5 =

    0.1268   -0.0850   -0.0056
   -0.0850    0.1188    0.0507
   -0.0028    0.0254    0.0529

>> S6 = Sbar(S, -40)

S6 =

    0.1188   -0.0850    0.0507
   -0.0850    0.1268   -0.0056
    0.0254   -0.0028    0.0529

>> S7 = Sbar(S, -30)

S7 =

    0.0932   -0.0669    0.0912
   -0.0669    0.1161   -0.0515
    0.0456   -0.0258    0.0892

>> S8 = Sbar(S, -20)

S8 =

    0.0594   -0.0391    0.0959
   -0.0391    0.0945   -0.0664
    0.0479   -0.0332    0.1447
>> S9 = Sbar(S, -10)
```

```
S9 =

    0.0310   -0.0147    0.0608
   -0.0147    0.0740   -0.0451
    0.0304   -0.0226    0.1935

>> S10 = Sbar(S, 0)

S10 =

    0.0200   -0.0051         0
   -0.0051    0.0658         0
         0         0    0.2128

>> S11 = Sbar(S, 10)

S11 =

    0.0310   -0.0147   -0.0608
   -0.0147    0.0740    0.0451
   -0.0304    0.0226    0.1935

>> S12 = Sbar(S, 20)

S12 =

    0.0594   -0.0391   -0.0959
   -0.0391    0.0945    0.0664
   -0.0479    0.0332    0.1447

>> S13 = Sbar(S, 30)

S13 =

    0.0932   -0.0669   -0.0912
   -0.0669    0.1161    0.0515
   -0.0456    0.0258    0.0892

>> S14 = Sbar(S, 40)

S14 =

    0.1188   -0.0850   -0.0507
   -0.0850    0.1268    0.0056
   -0.0254    0.0028    0.0529
```

```
>> S15 = Sbar(S, 50)

S15 =

    0.1268   -0.0850    0.0056
   -0.0850    0.1188   -0.0507
    0.0028   -0.0254    0.0529

>> S16 = Sbar(S, 60)

S16 =

    0.1161   -0.0669    0.0515
   -0.0669    0.0932   -0.0912
    0.0258   -0.0456    0.0892

>> S17 = Sbar(S, 70)

S17 =

    0.0945   -0.0391    0.0664
   -0.0391    0.0594   -0.0959
    0.0332   -0.0479    0.1447

>> S18 = Sbar(S, 80)

S18 =

    0.0740   -0.0147    0.0451
   -0.0147    0.0310   -0.0608
    0.0226   -0.0304    0.1935

>> S19 = Sbar(S, 90)

S19 =

    0.0658   -0.0051    0.0000
   -0.0051    0.0200   -0.0000
    0.0000   -0.0000    0.2128

>> Etaxyx1 = Etaxyx(S1)

Etaxyx1 =

 -2.6414e-016

>> Etaxyx2 = Etaxyx(S2)
```

```
Etaxyx2 =

   -0.6095

>> Etaxyx3 = Etaxyx(S3)

Etaxyx3 =

   -0.7031

>> Etaxyx4 = Etaxyx(S4)

Etaxyx4 =

   -0.4437

>> Etaxyx5 = Etaxyx(S5)

Etaxyx5 =

   -0.0444

>> Etaxyx6 = Etaxyx(S6)

Etaxyx6 =

    0.4269

>> Etaxyx7 = Etaxyx(S7)

Etaxyx7 =

    0.9779

>> Etaxyx8 = Etaxyx(S8)

Etaxyx8 =

    1.6138

>> Etaxyx9 = Etaxyx(S9)

Etaxyx9 =

    1.9599

>> Etaxyx10 = Etaxyx(S10)
```

```
Etaxyx10 =

     0

>> Etaxyx11 = Etaxyx(S11)

Etaxyx11 =

   -1.9599

>> Etaxyx12 = Etaxyx(S12)

Etaxyx12 =

   -1.6138

>> Etaxyx13 = Etaxyx(S13)

Etaxyx13 =

   -0.9779

>> Etaxyx14 = Etaxyx(S14)

Etaxyx14 =

   -0.4269

>> Etaxyx15 = Etaxyx(S15)

Etaxyx15 =

    0.0444

>> Etaxyx16 = Etaxyx(S16)

Etaxyx16 =

    0.4437

>> Etaxyx17 = Etaxyx(S17)

Etaxyx17 =

    0.7031
```

```
>> Etaxyx18 = Etaxyx(S18)

Etaxyx18 =

    0.6095

>> Etaxyx19 = Etaxyx(S19)

Etaxyx19 =

  2.6414e-016

>> x = [-90 -80 -70 -60 -50 -40 -30 -20 -10 0 10 20 30 40 50 60 70
       80 90]

x =

   -90   -80   -70   -60   -50   -40   -30   -20   -10     0    10
   20    30    40    50    60    70    80    90

>> y1 = [Etaxyx1 Etaxyx2 Etaxyx3 Etaxyx4 Etaxyx5 Etaxyx6 Etaxyx7
        Etaxyx8 Etaxyx9 Etaxyx10 Etaxyx11 Etaxyx12 Etaxyx13 Etaxyx14
        Etaxyx15 Etaxyx16 Etaxyx17 Etaxyx18 Etaxyx19]

y1 =

  Columns 1 through 14

    -0.0000   -0.6095   -0.7031   -0.4437   -0.0444    0.4269
     0.9779   1.6138    1.9599   0   -1.9599   -1.6138
    -0.9779   -0.4269

  Columns 15 through 19

     0.0444    0.4437    0.7031    0.6095    0.0000

>> plot(x,y1)
>> xlabel('\theta {degrees}');
>> ylabel('\eta_{xy,x}');
```

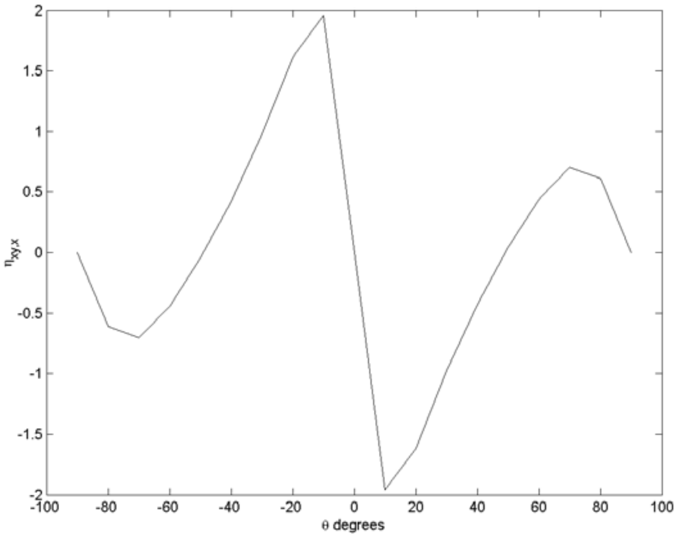

Fig. Variation of $\eta_{xy,\,x}$ versus θ for Problem 6.9

```
>> Etaxyy1 = Etaxyy(S1)

Etaxyy1 =

  1.1492e-015

>> Etaxyy2 = Etaxyy(S2)

Etaxyy2 =

    1.9599

>> Etaxyy3 = Etaxyy(S3)

Etaxyy3 =

    1.6138

>> Etaxyy4 = Etaxyy(S4)

Etaxyy4 =

    0.9779
```

```
>> Etaxyy5 = Etaxyy(S5)

Etaxyy5 =

    0.4269

>> Etaxyy6 = Etaxyy(S6)

Etaxyy6 =

   -0.0444

>> Etaxyy7 = Etaxyy(S7)

Etaxyy7 =

   -0.4437

>> Etaxyy8 = Etaxyy(S8)

Etaxyy8 =

   -0.7031

>> Etaxyy9 = Etaxyy(S9)

Etaxyy9 =

   -0.6095

>> Etaxyy10 = Etaxyy(S10)

Etaxyy10 =

     0

>> Etaxyy11 = Etaxyy(S11)

Etaxyy11 =

    0.6095

>> Etaxyy12 = Etaxyy(S12)

Etaxyy12 =

    0.7031
```

```
>> Etaxyy13 = Etaxyy(S13)

Etaxyy13 =

    0.4437

>> Etaxyy14 = Etaxyy(S14)

Etaxyy14 =

    0.0444

>> Etaxyy15 = Etaxyy(S15)


Etaxyy15 =

   -0.4269

>> Etaxyy16 = Etaxyy(S16)

Etaxyy16 =

   -0.9779

>> Etaxyy17 = Etaxyy(S17)

Etaxyy17 =

   -1.6138

>> Etaxyy18 = Etaxyy(S18)

Etaxyy18 =

   -1.9599

>> Etaxyy19 = Etaxyy(S19)

Etaxyy19 =

 -1.1492e-015

>> y2 = [Etaxyy1 Etaxyy2 Etaxyy3 Etaxyy4 Etaxyy5 Etaxyy6 Etaxyy7
        Etaxyy8 Etaxyy9 Etaxyy10 Etaxyy11 Etaxyy12 Etaxyy13 Etaxyy14
        Etaxyy15 Etaxyy16 Etaxyy17 Etaxyy18 Etaxyy19]
```

```
y2 =

  Columns 1 through 14

     0.0000    1.9599    1.6138    0.9779    0.4269   -0.0444
    -0.4437   -0.7031   -0.6095   0     0.6095    0.7031
     0.4437    0.0444

  Columns 15 through 19

    -0.4269   -0.9779   -1.6138   -1.9599   -0.0000

>> plot(x,y2)
>> xlabel('\theta {degrees}');
>> ylabel('\eta_{xy,y}');
```

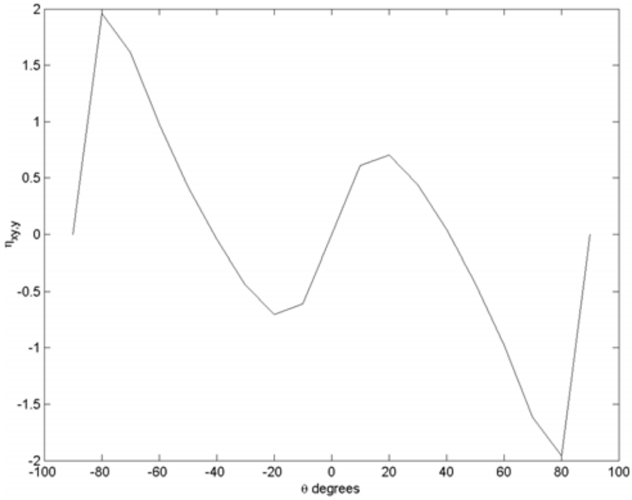

Fig. Variation of $\eta_{xy,\,y}$ versus θ for Problem 6.9

Problem 6.10

Continuing with the commands from Problem 6.9, we obtain:

```
>> Etaxxy1 = Etaxxy(S1)

Etaxxy1 =

 -8.1673e-017
```

```
>> Etaxxy2 = Etaxxy(S2)

Etaxxy2 =

   -0.2333

>> Etaxxy3 = Etaxxy(S3)

Etaxxy3 =

   -0.4591

>> Etaxxy4 = Etaxxy(S4)

Etaxxy4 =

   -0.5779

>> Etaxxy5 = Etaxxy(S5)

Etaxxy5 =

   -0.1064

>> Etaxxy6 = Etaxxy(S6)

Etaxxy6 =

    0.9581

>> Etaxxy7 = Etaxxy(S7)

Etaxxy7 =

    1.0226

>> Etaxxy8 = Etaxxy(S8)

Etaxxy8 =

    0.6626

>> Etaxxy9 = Etaxxy(S9)

Etaxxy9 =

    0.3142
```

```
>> Etaxxy10 = Etaxxy(S10)

Etaxxy10 =

     0

>> Etaxxy11 = Etaxxy(S11)

Etaxxy11 =

   -0.3142

>> Etaxxy12 = Etaxxy(S12)

Etaxxy12 =

   -0.6626

>> Etaxxy13 = Etaxxy(S13)

Etaxxy13 =

   -1.0226

>> Etaxxy14 = Etaxxy(S14)

Etaxxy14 =

   -0.9581

>> Etaxxy15 = Etaxxy(S15)

Etaxxy15 =

    0.1064

>> Etaxxy16 = Etaxxy(S16)

Etaxxy16 =

    0.5779

>> Etaxxy17 = Etaxxy(S17)

Etaxxy17 =

    0.4591
```

```
>> Etaxxy18 = Etaxxy(S18)

Etaxxy18 =

    0.2333

>> Etaxxy19 = Etaxxy(S19)

Etaxxy19 =

  8.1673e-017

>> y3 = [Etaxxy1 Etaxxy2 Etaxxy3 Etaxxy4 Etaxxy5 Etaxxy6 Etaxxy7
        Etaxxy8 Etaxxy9 Etaxxy10 Etaxxy11 Etaxxy12 Etaxxy13 Etaxxy14
        Etaxxy15 Etaxxy16 Etaxxy17 Etaxxy18 Etaxxy19]

y3 =

  Columns 1 through 14

    -0.0000   -0.2333   -0.4591   -0.5779   -0.1064    0.9581
     1.0226    0.6626    0.3142         0   -0.3142   -0.6626
    -1.0226   -0.9581

  Columns 15 through 19

     0.1064    0.5779    0.4591    0.2333    0.0000

>> plot(x,y3)
>> xlabel('\theta {degrees}');
>> ylabel('\eta_{x,xy}');

>> Etayxy1 = Etayxy(S1)

Etayxy1 =

  1.0803e-016

>> Etayxy2 = Etayxy(S2)

Etayxy2 =

    0.3142

>> Etayxy3 = Etayxy(S3)

Etayxy3 =

    0.6626
```

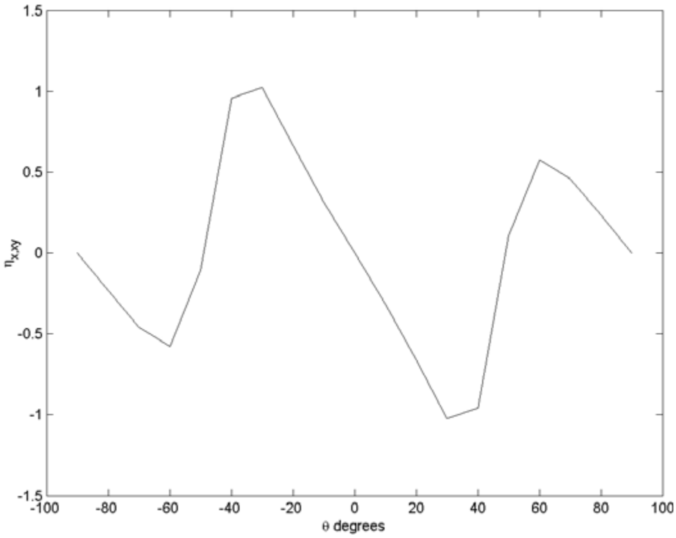

Fig. Variation of $\eta_{x,\,xy}$ versus θ for Problem 6.10

```
>> Etayxy4 = Etayxy(S4)

Etayxy4 =

    1.0226

>> Etayxy5 = Etayxy(S5)

Etayxy5 =

    0.9581

>> Etayxy6 = Etayxy(S6)

Etayxy6 =

   -0.1064

>> Etayxy7 = Etayxy(S7)

Etayxy7 =

   -0.5779
```

```
>> Etayxy8 = Etayxy(S8)

Etayxy8 =

   -0.4591

>> Etayxy9 = Etayxy(S9)

Etayxy9 =

   -0.2333

>> Etayxy10 = Etayxy(S10)

Etayxy10 =

     0

>> Etayxy11 = Etayxy(S11)

Etayxy11 =

    0.2333

>> Etayxy12 = Etayxy(S12)

Etayxy12 =

    0.4591

>> Etayxy13 = Etayxy(S13)

Etayxy13 =

    0.5779

>> Etayxy14 = Etayxy(S14)

Etayxy14 =

    0.1064

>> Etayxy15 = Etayxy(S15)

Etayxy15 =

   -0.9581
```

```
>> Etayxy16 = Etayxy(S16)

Etayxy16 =

   -1.0226

>> Etayxy17 = Etayxy(S17)

Etayxy17 =

   -0.6626

>> Etayxy18 = Etayxy(S18)

Etayxy18 =

   -0.3142

>> Etayxy19 = Etayxy(S19)

Etayxy19 =

   -1.0803e-016

>> y4 = [Etayxy1 Etayxy2 Etayxy3 Etayxy4 Etayxy5 Etayxy6 Etayxy7
        Etayxy8 Etayxy9 Etayxy10 Etayxy11 Etayxy12 Etayxy13 Etayxy14
        Etayxy15 Etayxy16 Etayxy17 Etayxy18 Etayxy19]

y4 =

  Columns 1 through 14

     0.0000    0.3142    0.6626    1.0226    0.9581   -0.1064
    -0.5779   -0.4591   -0.2333         0    0.2333    0.4591
     0.5779    0.1064

  Columns 15 through 19

    -0.9581   -1.0226   -0.6626   -0.3142   -0.0000

>> plot(x,y4)
>> xlabel('\theta {degrees}');
>> ylabel('\eta_{y,xy}');
```

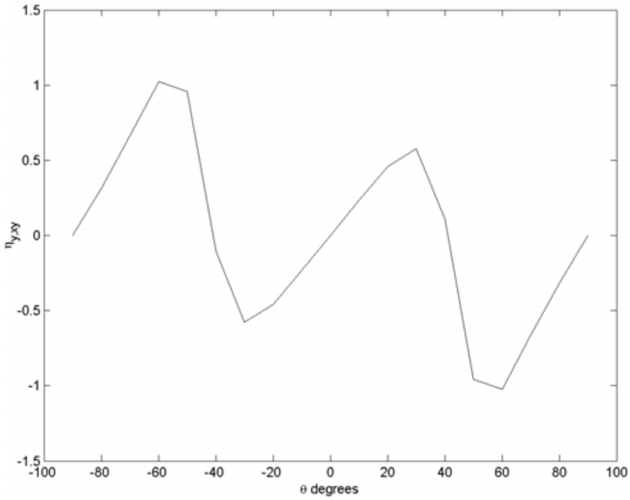

Fig. Variation of $\eta_{y,\,xy}$ versus θ for Problem 6.10

Problem 7.1

```
EDU>> epsilon1 = Strains(500e-6,0,0,0,0,0,-0.300)

epsilon1 =

  1.0e-003 *

    0.5000
         0
         0

EDU>> epsilon2 = Strains(500e-6,0,0,0,0,0,-0.150)

epsilon2 =

  1.0e-003 *

    0.5000
         0
         0

EDU>> epsilon3 = Strains(500e-6,0,0,0,0,0,0)

epsilon3 =

  1.0e-003 *
```

```
    0.5000
         0
         0

EDU>> epsilon4 = Strains(500e-6,0,0,0,0,0,0.150)

epsilon4 =

  1.0e-003 *

    0.5000
         0
         0

EDU>> epsilon5 = Strains(500e-6,0,0,0,0,0,0.300)

epsilon5 =

  1.0e-003 *

    0.5000
         0
         0

EDU>> Q = ReducedStiffness(50.0, 15.2, 0.254, 4.70)

Q =

   51.0003    3.9380         0
    3.9380   15.5041         0
         0         0    4.7000

EDU>> Qbar1 = Qbar(Q,0)

Qbar1 =

   51.0003    3.9380         0
    3.9380   15.5041         0
         0         0    4.7000

EDU>> Qbar2 = Qbar(Q,90)

Qbar2 =

   15.5041    3.9380   -0.0000
    3.9380   51.0003    0.0000
   -0.0000    0.0000    4.7000
```

```
EDU>> Qbar3 = Qbar(Q,90)

Qbar3 =

   15.5041    3.9380   -0.0000
    3.9380   51.0003    0.0000
   -0.0000    0.0000    4.7000

EDU>> Qbar4 = Qbar(Q,0)

Qbar4 =

   51.0003    3.9380         0
    3.9380   15.5041         0
         0         0    4.7000

EDU>> sigma1a = Qbar1*epsilon1*1e3

sigma1a =

   25.5001
    1.9690
         0

EDU>> sigma1b = Qbar1*epsilon2*1e3

sigma1b =

   25.5001
    1.9690
         0

EDU>> sigma2a = Qbar2*epsilon2*1e3

sigma2a =

    7.7520
    1.9690
   -0.0000

EDU>> sigma2b = Qbar2*epsilon3*1e3

sigma2b =

    7.7520
    1.9690
   -0.0000
```

```
EDU>> sigma3a = Qbar3*epsilon3*1e3

sigma3a =

    7.7520
    1.9690
   -0.0000

EDU>> sigma3b = Qbar3*epsilon4*1e3

sigma3b =

    7.7520
    1.9690
   -0.0000


EDU>> sigma4a = Qbar4*epsilon4*1e3

sigma4a =

   25.5001
    1.9690
         0

EDU>> sigma4b = Qbar4*epsilon5*1e3

sigma4b =

   25.5001
    1.9690
         0

EDU>> y = [0.300 0.150 0.150 0 0 -0.150 -0.150 -0.300]

y =

    0.3000    0.1500    0.1500    0    0   -0.1500
   -0.1500   -0.3000

EDU>> x = [sigma4b(1) sigma4a(1) sigma3b(1) sigma3a(1) sigma2b(1)
          sigma2a(1) sigma1b(1) sigma1a(1)]

x =

   25.5001   25.5001    7.7520    7.7520    7.7520    7.7520
   25.5001   25.5001
```

```
EDU>> plot(x,y)
EDU>> xlabel('\sigma_x (MPa)')
EDU>> ylabel('z (mm)')
```

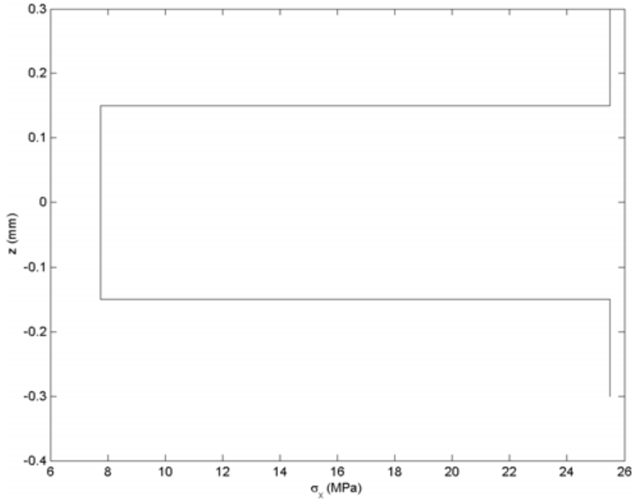

Fig. Variation of σ_x versus z for Problem 7.1

```
EDU>> x = [sigma4b(2) sigma4a(2) sigma3b(2) sigma3a(2) sigma2b(2)
          sigma2a(2) sigma1b(2) sigma1a(2)]

x =

    1.9690    1.9690    1.9690    1.9690    1.9690    1.9690
    1.9690    1.9690

EDU>> plot(x,y)
EDU>> ylabel('z (mm)')
EDU>> xlabel('\sigma_y (MPa)')

EDU>> x = [sigma4b(3) sigma4a(3) sigma3b(3) sigma3a(3) sigma2b(3)
          sigma2a(3) sigma1b(3) sigma1a(3)]

x =

  1.0e-015 *

         0    0   -0.2102   -0.2102   -0.2102   -0.2102
         0    0
```

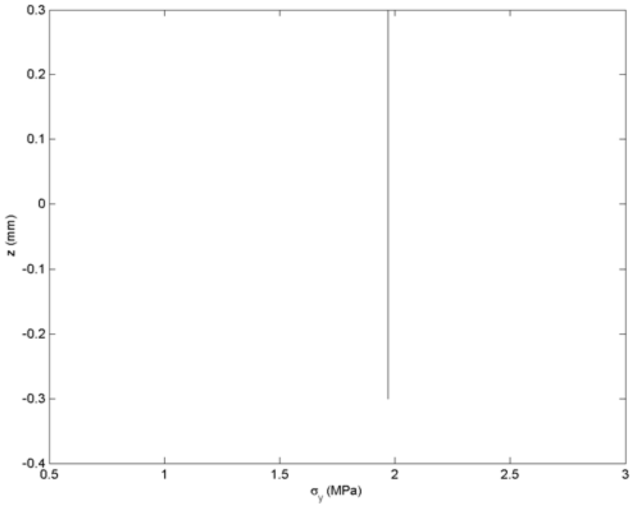

Fig. Variation of σ_y versus z for Problem 7.1

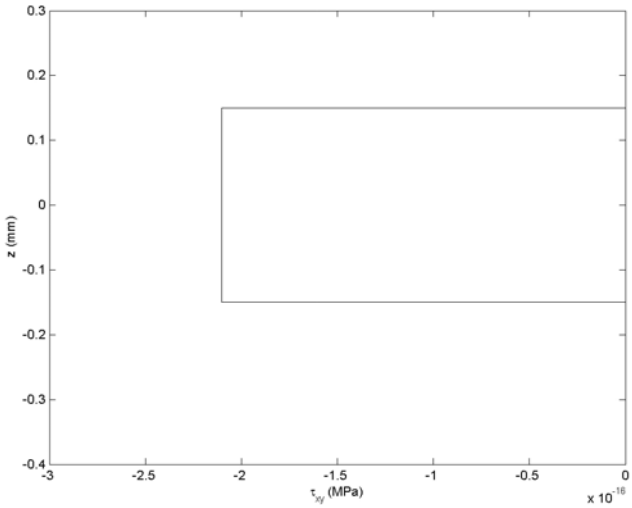

Fig. Variation of τ_{xy} versus z for Problem 7.1

```
EDU>> plot(x,y)
EDU>> ylabel('z (mm)')
EDU>> xlabel('\tau_{xy} (MPa)')

EDU>> Nx = 0.150e-3 * (sigma1a(1) + sigma2a(1) + sigma3a(1) +
         sigma4a(1))
```

```
Nx =

    0.0100

EDU>> Ny = 0.150e-3 * (sigma1a(2) + sigma2a(2) + sigma3a(2) +
           sigma4a(2))

Ny =

    0.0012

EDU>> Nxy = 0.150e-3 * (sigma1a(3) + sigma2a(3) + sigma3a(3) +
            sigma4a(3))

Nxy =

 -6.3064e-020

EDU>> Mx = sigma1a(1)*((-0.150e-3)^2 - (0.300e-3)^2) + sigma2a(1)*(0 -
           (-0.150e-3)^2) + sigma3a(1)*((0.150e-3)^2 - 0) +
           sigma4a(1)*((0.300e-3)^2 - (0.150e-3)^2)
Mx =

     0

EDU>> My = sigma1a(2)*((-0.150e-3)^2 - (0.300e-3)^2) + sigma2a(2)*(0
           - (-0.150e-3)^2) + sigma3a(2)*((0.150e-3)^2 - 0) +
           sigma4a(2)*((0.300e-3)^2 - (0.150e-3)^2)

My =

     0

EDU>> Mxy = sigma1a(3)*((-0.150e-3)^2 - (0.300e-3)^2) + sigma2a(3)*(0
            - (-0.150e-3)^2) + sigma3a(3)*((0.150e-3)^2 - 0) +
            sigma4a(3)*((0.300e-3)^2 - (0.150e-3)^2)

Mxy =

     0

EDU>> T1 = T(0)

T1 =

     1     0     0
     0     1     0
     0     0     1
```

```
EDU>> T2 = T(90)

T2 =

    0.0000    1.0000    0.0000
    1.0000    0.0000   -0.0000
   -0.0000    0.0000   -1.0000

EDU>> T3 = T(90)

T3 =

    0.0000    1.0000    0.0000
    1.0000    0.0000   -0.0000
   -0.0000    0.0000   -1.0000

EDU>> T4 = T(0)

T4 =

     1     0     0
     0     1     0
     0     0     1

EDU>> eps1a = T1*epsilon1

eps1a =

  1.0e-003 *

    0.5000
         0
         0

EDU>> eps1b = T1*epsilon2

eps1b =

  1.0e-003 *

    0.5000
         0
         0

EDU>> eps2a = T2*epsilon2
```

```
eps2a =

  1.0e-003 *

    0.0000
    0.5000
   -0.0000

EDU>> eps2b = T2*epsilon3

eps2b =

  1.0e-003 *

    0.0000
    0.5000
   -0.0000

EDU>> eps3a = T3*epsilon3

eps3a =

  1.0e-003 *

    0.0000
    0.5000
   -0.0000

EDU>> eps3b = T3*epsilon4

eps3b =

  1.0e-003 *

    0.0000
    0.5000
   -0.0000

EDU>> eps4a = T4*epsilon4

eps4a =

  1.0e-003 *

    0.5000
         0
         0
```

```
EDU>> eps4b = T4*epsilon5

eps4b =

  1.0e-003 *

    0.5000
         0
         0

EDU>> sig1 = T1*sigma1a

sig1 =

   25.5001
    1.9690
         0

EDU>> sig2 = T2*sigma2a

sig2 =

    1.9690
    7.7520
   -0.0000

EDU>> sig3 = T3*sigma3a

sig3 =

    1.9690
    7.7520
   -0.0000

EDU>> sig4 = T4*sigma4a

sig4 =

   25.5001
    1.9690
         0
```

References

Books on Mechanics of Composite Materials

1. Hyer, M., Stress Analysis of Fiber-Reinforced Composite Materials, McGraw-Hill, 1998.
2. Daniel, I. and Ishai, O., Engineering Mechanics of Composite Materials, Oxford University Press, 1994.
3. Gibson, R., Principles of Composite Material Mechanics, McGraw-Hill, 1994.
4. Hoskin, B. and Baker, A. (editors), Composite Materials for Aircraft Structures, American Institute of Aeronautics and Astronautics, 1986.
5. Eckold, G., Design and Manufacture of Composite Structures, McGraw-Hill, 1994.
6. Swanson, S., Introduction to Design and Analysis with Advanced Composite Materials, Prentice Hall, 1997.
7. Herakovich, C., Mechanics of Fibrous Composites, John Wiley & Sons, 1998.
8. Jones, R., Mechanics of Composite Materials, Second Edition, Taylor and Francis, 1998.
9. Christensen, R., Mechanics of Composite Materials, Krieger Publishing Company, 1991.
10. Kaw, A., Mechanics of Composite Materials, CRC,Press 1997.
11. Vasiliev, V. and Morozov, E., Mechanics and Analysis of Composite Materials, Elsevier Science, 2001.
12. Soares, C. and Freitas, M., Mechanics of Composite Materials and Structures, Kluwer Academic Publishing, 1999.

MATLAB Books

13. Etter, D., Engineering Problem Solving with MATLAB, Second Edition, Prentice Hall, 1996.
14. Part-Enander, E. and Sjoberg, A., The MATLAB 5 Handbook, Addison-Wesley, 1999.
15. Biran, A. and Breiner, M., MATLAB 5 for Engineers, Addison-Wesley, 1999.
16. Higham, D. and Higham, N., MATLAB Guide, SIAM, 2000.

17. Hanselman, D. and Littlefield, B., Mastering MATLAB 6, Prentice Hall, 2000.
18. Palm, W., Introduction to MATLAB 6 for Engineers, McGraw-Hill, 2000.
19. Chen, K., Giblin, P. and Irving, A., Mathematical Explorations with MATLAB, Cambridge University Press, 1999.
20. Polking, J. and Arnold, D., Ordinary Differential Equations Using MATLAB, Second Edition, Prentice Hall, 1999.
21. Fausett, L., Applied Numerical Analysis Using MATLAB, Prentice Hall, 1999.

MATLAB Tutorials on the Internet

22. Cavers, I., An Introductory Guide to MATLAB, Second Edition, 1998, http://www.cs.ubc.ca/spider/cavers/MatlabGuide/guide.html
23. Gockenback, M., A Practical Introduction to MATLAB: Updated for MATLAB 5, http://www.math.mtu.edu/~msgocken/intro/intro.html
24. Huber, T., MATLAB Tutorials: Envision-It! Workshop 1997, http://physics.gac.edu/~huber/envision/tutor2/
25. Maneval, J., Helpful Information for Using MATLAB, http://www.facstaff.bucknell.edu/maneval/help211/helpmain.html
26. Recktenwald, G., MATLAB Hypertext Reference, 1996, http://www.me.pdx.edu/~gerry/MATLAB/
27. University of Texas, MATLAB Online Reference Documentation, http://www.utexas.edu/cc/math/Matlab/Manual/ReferenceTOC.html
28. Southern Illinois University at Carbondale, MATLAB Tutorials, http://www.math.siu.edu/matlab/tutorials.html
29. MATLAB Educational Sites, http://www.eece.maine.edu/mm/matweb.html

Contents of the Accompanying CD-ROM

The accompanying CD-ROM includes two folders as follows:

1. *M-Files.* This folder includes the 44 MATLAB functions written specifically to be used with this book. In order to use them they should be copied to the working directory in your MATLAB folder on the hard disk or you can set the MATLAB path to the correct folder that includes these files.
2. Solutions to most of the problems in the book. Specifically, detailed solutions are included to all the problem of the first six chapters.

Index