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:

$$\begin{Bmatrix} N_x \\ N_y \\ N_{xy} \end{Bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{16} \\ A_{12} & A_{22} & A_{26} \\ A_{16} & A_{26} & A_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_x^0 \\ \varepsilon_y^0 \\ \gamma_{xy}^0 \end{Bmatrix} + \begin{bmatrix} B_{11} & B_{12} & B_{16} \\ B_{12} & B_{22} & B_{26} \\ B_{16} & B_{26} & B_{66} \end{bmatrix} \begin{Bmatrix} \kappa_x^0 \\ \kappa_y^0 \\ \kappa_{XY}^0 \end{Bmatrix} \tag{8.1}$$

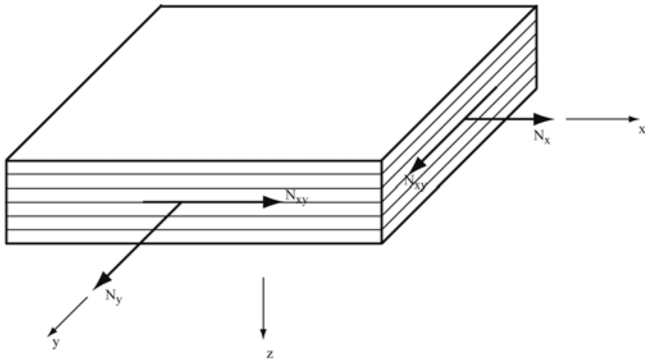

Fig. 8.1. Schematic illustration of the force resultants on a composite laminate

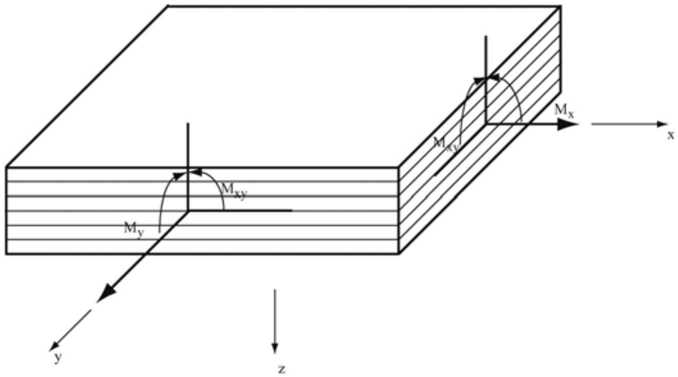

Fig. 8.2. Schematic illustration of the moment resultants on a composite laminate

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:

$$N_x = \int_{-H/2}^{H/2} \left[\bar{Q}_{11}\left(\varepsilon_x^0 + z\kappa_x^0\right) + \bar{Q}_{12}\left(\varepsilon_y^0 + z\kappa_y^0\right) + \bar{Q}_{16}\left(\gamma_{xy}^0 + z\kappa_{xy}^0\right)\right] dz \tag{8.14}$$

Expanding (8.14), we obtain:

$$\begin{aligned} N_x = {} & \varepsilon_x^0 \int_{-H/2}^{H/2} \bar{Q}_{11}\, dz + \kappa_x^0 \int_{-H/2}^{H/2} \bar{Q}_{11}\, zdz + \varepsilon_y^0 \int_{-H/2}^{H/2} \bar{Q}_{12}\, dz + \kappa_y^0 \int_{-H/2}^{H/2} \bar{Q}_{12}\, zdz \\ & + \gamma_{xy}^0 \int_{-H/2}^{H/2} \bar{Q}_{16}\, dz + \kappa_{xy}^0 \int_{-H/2}^{H/2} \bar{Q}_{16}\, zdz \end{aligned} \tag{8.15}$$

Next, we expand the first term of (8.15) as follows:

$$\int_{-H/2}^{H/2} \bar{Q}_{11}\, dz = \int_{z_0}^{z_1} \bar{Q}_{11} dz + \int_{z_1}^{z_2} \bar{Q}_{11} dz + \cdots + \int_{z_{k-1}}^{z_k} \bar{Q}_{11} dz + \cdots + \int_{z_{N-1}}^{z_N} \bar{Q}_{11} dz \tag{8.16}$$

Recognizing that $\bar{Q}_{11}$ is constant within each layer, it can be taken outside the integrals above leading to the following expression:

$$\begin{aligned} \int_{-H/2}^{H/2} \bar{Q}_{11}\, dz = {} & \bar{Q}_{11}\left(z_1 - z_0\right) + \bar{Q}_{11}\left(z_2 - z_1\right) + \cdots + \bar{Q}_{11}\left(z_k - z_{k-1}\right) \\ & + \cdots + \bar{Q}_{11}\left(z_N - z_{N-1}\right) \end{aligned} \tag{8.17}$$

The above equation can be re-written as follows:

$$\int_{-H/2}^{H/2} \bar{Q}_{11}\, dz = \sum_{k=1}^{N} \bar{Q}_{11}\left(z_k - z_{k-1}\right) = A_{11} \tag{8.18}$$

Similarly, we can show that the other five integrals of (8.15) can be written as follows:

$$\int_{-H/2}^{H/2} \bar{Q}_{12}\, dz = \sum_{k=1}^{N} \bar{Q}_{12}\left(z_k - z_{k-1}\right) = A_{12} \tag{8.19a}$$

$$\int_{-H/2}^{H/2} \bar{Q}_{16}\, dz = \sum_{k=1}^{N} \bar{Q}_{16}\left(z_k - z_{k-1}\right) = A_{16} \tag{8.19b}$$

$$\int_{-H/2}^{H/2} \bar{Q}_{11}\, zdz = \frac{1}{2} \sum_{k=1}^{N} \bar{Q}_{11}\left(z_k^2 - z_{k-1}^2\right) = B_{11} \tag{8.19c}$$

$$\int_{-H/2}^{H/2} \bar{Q}_{12}\, z dz = \frac{1}{2}\sum_{k=1}^{N} \bar{Q}_{12}\left(z_k^2 - z_{k-1}^2\right) = B_{12} \tag{8.19d}$$

$$\int_{-H/2}^{H/2} \bar{Q}_{16}\, z dz = \frac{1}{2}\sum_{k=1}^{N} \bar{Q}_{16}\left(z_k^2 - z_{k-1}^2\right) = B_{16} \tag{8.19e}$$

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.