Chapter 3

Linear Systems and Matrices

Application 3.2
Automated Row Operations

Computer algebra systems are often used to ease the labor of matrix computations, including elementary row operations. For instance, suppose we want to reduce the 3×4 augmented coefficient matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 1 & 4 \\ 3 & 8 & 7 & 20 \\ 2 & 7 & 9 & 23 \end{bmatrix} \tag{1}$$

of Example 3 in the text to an echelon matrix. As illustrated below we can use *Maple*, *Mathematica*, or MATLAB to carry out the successive row operations interactively — each with a one-liner that displays the intermediate result for our inspection to determine the next operation to perform. You should verify that these operations (using your computer algebra system of choice) yield the echelon matrix

$$\mathbf{E} = \begin{bmatrix} 1 & 2 & 1 & 4 \\ 0 & 1 & 2 & 4 \\ 0 & 0 & 1 & 3 \end{bmatrix}. \tag{2}$$

Then you can use "automated" elementary row operations to find the echelon matrices requested in Problems 11–22 of this section in the text.

Using *Maple*

The 3×4 matrix in (1) can be entered with the *Maple* command

```
with(linalg):
A := array( [[1, 2, 1,  4],
             [3, 8, 7, 20],
             [2, 7, 9, 23]] );
```

$$A := \begin{bmatrix} 1 & 2 & 1 & 4 \\ 3 & 8 & 7 & 20 \\ 2 & 7 & 9 & 23 \end{bmatrix}$$

The *Maple* **linalg** package has built-in commands that can be used to carry out the reduction of **A**, one row operation at a time, as follows:

```
A := addrow(A,1,2,-3);          # (-3)R1 + R2
```

$$A := \begin{bmatrix} 1 & 2 & 1 & 4 \\ 0 & 2 & 4 & 8 \\ 2 & 7 & 9 & 23 \end{bmatrix}$$

```
A := addrow(A,1,3,-2);          # (-2)R1 + R3
```

$$A := \begin{bmatrix} 1 & 2 & 1 & 4 \\ 0 & 2 & 4 & 8 \\ 0 & 3 & 7 & 15 \end{bmatrix}$$

```
A := mulrow(A,2,1/2);           # (1/2)R2
```

$$A := \begin{bmatrix} 1 & 2 & 1 & 4 \\ 0 & 1 & 2 & 4 \\ 0 & 3 & 7 & 15 \end{bmatrix}$$

```
A := addrow(A,2,3,-3);          # (-3)R2 + R3
```

$$A := \begin{bmatrix} 1 & 2 & 1 & 4 \\ 0 & 1 & 2 & 4 \\ 0 & 0 & 1 & 3 \end{bmatrix}$$

The inserted remarks using the notation of Fig. 3.5 should make clear the structure of these row operations in *Maple*. Thus the command **addrow(A,p,q,c)** adds c times row p to row q, and **mulrow(A,2,1/2)** multiplies row p by c. If you wanted to interchange the 1st and 3rd rows — and proceed to solve the corresponding system by "forward substitution" rather than back substitution — this could be done as follows:

```
A := swaprow(A,1,3);          # Swap(R1,R3)
```

$$A := \begin{bmatrix} 0 & 0 & 1 & 3 \\ 0 & 1 & 2 & 4 \\ 1 & 2 & 1 & 4 \end{bmatrix}$$

Using *Mathematica*

The 3×4 matrix in (1) can be entered with the *Mathematica* command

```
A = {{1, 2, 1,  4},
     {3, 8, 7, 20},
     {2, 7, 9, 23}}
```

$$\begin{pmatrix} 1 & 2 & 1 & 4 \\ 3 & 8 & 7 & 20 \\ 2 & 7 & 9 & 23 \end{pmatrix}$$

In *Mathematica* the *i*th row of the matrix **A** is denoted by `A[[i]]`, and we can operate directly on rows, one at a time, to carry out the desired reduction to echelon form:

```
A[[2]] = (-3)A[[1]] + A[[2]];    (*  (-3)R1 + R2  *)
A
```

$$\begin{pmatrix} 1 & 2 & 1 & 4 \\ 0 & 2 & 4 & 8 \\ 2 & 7 & 9 & 23 \end{pmatrix}$$

```
A[[3]] = (-2)A[[1]] + A[[3]];    (*  (-2)R1 + R3  *)
A
```

$$\begin{pmatrix} 1 & 2 & 1 & 4 \\ 0 & 2 & 4 & 8 \\ 0 & 3 & 7 & 15 \end{pmatrix}$$

```
A[[2]] = (1/2)A[[2]];           (*   (1/2)R2     *)
A
```

$$\begin{pmatrix} 1 & 2 & 1 & 4 \\ 0 & 1 & 2 & 4 \\ 0 & 3 & 7 & 15 \end{pmatrix}$$

```
A[[3]] = (-3)A[[2]] + A[[3]];    (*  (-3)R2 + R3  *)
A
```

$$\begin{pmatrix} 1 & 2 & 1 & 4 \\ 0 & 1 & 2 & 4 \\ 0 & 0 & 1 & 3 \end{pmatrix}$$

The inserted remarks using the notation of Fig. 3.5 should make clear the structure of these row operations in *Mathematica*. Each redefines — as we specify — a single row of the matrix **A**. If you wanted to interchange the 1st and 3rd rows — and proceed to solve the corresponding system by "forward substitution" rather than back substitution — this could be done as follows:

```
A[[{1,3}]] = A[[{3,1}]];      (*   Swap(R1,R3)   *)
A
```

$$\begin{pmatrix} 0 & 0 & 1 & 3 \\ 0 & 1 & 2 & 4 \\ 1 & 2 & 1 & 4 \end{pmatrix}$$

Here, **`A[[{1,3}]]`** specifies the submatrix of the matrix **A** consisting of its 1st and 3rd rows.

Using MATLAB

The 3×4 matrix in (1) can be entered with the MATLAB command

```
A = [1  2  1   4
     3  8  7  20
     2  7  9  23]

A =
     1     2     1     4
     3     8     7    20
     2     7     9    23
```

In MATLAB the *i*th row of the matrix **A** is denoted by **`A(i,:)`**, and we can operate directly on rows, one at a time, to carry out the desired reduction to echelon form:

```
A(2,:) = (-3)*A(1,:) + A(2,:)       % (-3)R1 + R2

A =
     1     2     1     4
     0     2     4     8
     2     7     9    23
```

```
A(3,:) = (-2)*A(1,:) + A(3,:)       % (-2)R1 + R2

A =
     1     2     1     4
     0     2     4     8
     0     3     7    15

A(2,:) = (1/2)*A(2,:)               % (1/2)R2

A =
     1     2     1     4
     0     1     2     4
     0     3     7    15

A(3,:) = (-3)*A(2,:) + A(3,:)       % (-3)R2 + R3

A =

     1     2     1     4
     0     1     2     4
     0     0     1     3
```

The inserted remarks using the notation of Fig. 3.5 should make clear the structure of these row operations in MATLAB. Each redefines — as we specify — a single row of the matrix **A**. If you wanted to interchange the 1st and 3rd rows — and proceed to solve the corresponding system by "forward substitution" rather than back substitution — this could be done as follows:

```
A([1 3],:) = A([3 1],:)             % Swap(R1,R3)

A =
     0     0     1     3
     0     1     2     4
     1     2     1     4
```

Here, **`A([1 3],:)`** specifies the submatrix of the matrix **A** consisting of its 1st and 3rd rows.