```
>> B(12:15)='Bill'
B =
My name is Bill Smith
>>
```

Using a colon to assign new characters to elements 12 through 15 in the vector `B`.

Strings can also be placed in a matrix. As with numbers, this is done by typing a semicolon ; (or pressing the **Enter** key) at the end of each row. Each row must be typed as a string, which means that it must be enclosed in single quotes. In addition, as with a numerical matrix, the number of elements in all the rows must be the same. This requirement can cause problems when the intention is to create rows with specific wording. Rows can be made to have the same number of elements by adding spaces.

MATLAB has a built-in function named `char` that creates an array with rows that have the same number of characters from an input of rows that are not of the same length. MATLAB makes the length of all the rows equal to the longest row by adding spaces at the end of the short lines. In the `char` function, the rows are entered as strings separated by a comma according to the following format:

```
variable_ name = char('string 1','string 2','string 3')
```

For example:

```
>> Info=char('Student Name:','John Smith','Grade:','A+')
Info =
Student Name:
John Smith
Grade:
A+
>>
```

A variable named `Info` is assigned four rows of strings, each with different length.

The function `char` creates an array with four rows with the same length as the longest row by adding empty spaces to the shorter lines.

A variable can be defined as a number or a string that is made up of the same digits. For example, as shown below, `x` is defined to be the number 536, and `y` is defined to be a string made up of the digits 536.

```
>> x=536
x =
   536
>> y='536'
y =
536
>>
```

The two variables are not the same even though they appear identical on the screen. The variable `x` can be used in mathematical expressions, while the variable `y` can not.

2.11 PROBLEMS

1. Create a row vector that has the elements: 6, $8 \cdot 3$, 81, $e^{2.5}$, $\sqrt{65}$, $\sin(\pi/3)$, and 23.05.

2. Create a column vector that has the elements: 44, 9, ln(51), 2^3, 0.1, and $5\tan(25°)$.

3. Create a row vector in which the first element is 0, and the last element is 42, with an increment of 3 between the elements (0, 3, 6, ……, 42).

4. Create a column vector in which the first element is 18, the elements decrease with increments of –4, and the last element is –22. (A column vector can be created by the transpose of a row vector.)

5. Create a row vector with 16 equally spaced elements in which the first element is 5 and the last element is 61.

6. Create a column vector with 14 equally spaced elements in which the first element is 3 and the last element is –36.

7. Using the colon symbol, create a row vector (assign it to a variable named `same`) with eleven elements that are all 4.

8. Create a vector, name it `Afirst`, that has 13 elements in which the first is 3, the increment is 4 and the last element is 51. Then, using the colon symbol, create a new vector, call it `Asecond`, that has seven elements. The first four elements are the first four elements of the vector `Afirst`, and the last three are the last three elements of the vector `Afirst`.

9. Create the matrix shown below by using the vector notation for creating vectors with constant spacing and/or the `linspace` command when entering the rows.

$$B = \begin{bmatrix} 0 & 4 & 8 & 12 & 16 & 20 & 24 & 28 \\ 69 & 68 & 67 & 66 & 65 & 64 & 63 & 62 \\ 1.4 & 1.1 & 0.8 & 0.5 & 0.2 & -0.1 & -0.4 & -0.7 \end{bmatrix}$$

10. Using the colon symbol, create a 3×5 matrix (assign it to a variable named `msame`) in which all the elements are the number 7.

11. Create three vectors:

$$a = [2\ -1\ 0\ 6],\quad b = [-5\ 20\ 12\ -3],\quad c = [10\ 7\ -2\ 1]$$

a) Use the three vectors in a MATLAB command to create a 3×4 matrix in which the rows are the vectors *a*, *b*, and *c*.

b) Use the three vectors in a MATLAB command to create a 4×3 matrix in which the columns are the vectors *a*, *b*, and *c*.

12. By hand (pencil and paper) write what will be displayed if the following commands are executed by MATLAB. Check your answers by executing the commands with MATLAB. (Parts *b*, *c*, and *d* use the vector that was defined in part *a*.)

a) `a=0:2:6` *b*) `b=[a a]` or `b=[a,a]`

c) `c=[a; a]`

d) `d=[a' a']` or `d=[a',a']`

13. The following vector is defined in MATLAB:

$$v = [2, 7, -3, 5, 0, 14, -1, 10, -6, 8]$$

By hand (pencil and paper) write what will be displayed if the following commands are executed by MATLAB. Check your answers by executing the commands with MATLAB.

a) `a=v(3:6)` *b*) `b=v([2,4:7,10])` *c*) `c=v([9,3,1,10])`

d) `d=[v([1,3 5]);v([2,4,6]);v([3,6,9])]`

14. Create the following matrix *A*: $A = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \end{bmatrix}$

Use the matrix *A* to:

a) Create a five-element row vector named `va` that contains the elements of the first row of *A*.

b) Create a three-element row vector named `vb` that contains the elements of the third column of *A*.

c) Create a eight-element row vector named `vc` that contains the elements of the second rows of *A* and the fourth column of *A*.

d) Create a six-element row vector named `vd` that contains the elements of the first and fifth columns of *A*.

15. Create the following matrix *B*: $B = \begin{bmatrix} 15 & 12 & 9 & 6 & 3 \\ 2 & 4 & 6 & 8 & 10 \\ 6 & 12 & 18 & 24 & 30 \end{bmatrix}$

Use the matrix *B* to:

a) Create a six-element column vector named `ua` that contains the elements of the second and fourth columns of *C*.

b) Create a five-element column vector named `ub` that contains the elements of the third row of *C*.

c) Create a nine-element row vector named `uc` that contains the elements of the second, fourth and fifth columns of *C*.

d) Create a eight-element row vector named `ud` that contains the elements of the first column and first row of *C*.

16. Create the following matrix *A*: $\begin{bmatrix} 0.1 & 0.2 & 0.3 & 0.4 & 0.5 & 0.6 & 0.7 \\ 14 & 12 & 10 & 8 & 6 & 4 & 2 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 3 & 6 & 9 & 12 & 15 & 18 & 21 \end{bmatrix}$

a) Create a 3×4 matrix *B* from the 1st, 2nd, and 3rd rows, and the 1st through the 4th columns of the matrix *A*.

b) Create a 2×7 matrix *C* from the 2nd, and 3rd rows, and all the columns of the matrix *A*.

17. The following matrix is defined in MATLAB: $M = \begin{bmatrix} 6 & 9 & 12 & 15 & 18 & 21 \\ 4 & 4 & 4 & 4 & 4 & 4 \\ 2 & 1 & 0 & -1 & -2 & -3 \\ -6 & -4 & -2 & 0 & 2 & 4 \end{bmatrix}$

By hand (pencil and paper) write what will be displayed if the following commands are executed by MATLAB. Check your answers by executing the commands with MATLAB.

a) `A=M([1,3],[2,4])`

b) `B=M(:,[1,4:6])`

c) `C=M([2,3],:)`

18. Using the `zeros`, `ones`, and `eye` commands create the following arrays:

a) $\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$ *b*) $\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$ *c*) $\begin{bmatrix} 1 & 1 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix}$

19. Use the `eye` command to create the array A shown on the left below, Then, using the colon to address the elements in the arrays, change A to be like the one shown on the right.

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

20. Create a vector $v = [1, 2, 3, \ldots, 35]$ with 35 elements. Then, use the reshape function to create a 5×7 matrix in which the first row is the numbers 1 2 3 4 5 6 7, the second row is the numbers 8 9 10 11 12 13 14, the third row is the numbers 15 through 21, and so on.

21. Create a 3×3 matrix A in which all the elements are 1. Then, reassign A to itself such that A will become:

$$A = \begin{bmatrix} 1&1&1&0&0&0 \\ 1&1&1&0&0&0 \\ 1&1&1&0&0&0 \\ 0&0&0&1&1&1 \\ 0&0&0&1&1&1 \\ 0&0&0&1&1&1 \end{bmatrix}$$

Chapter 3
Mathematical Operations with Arrays

Once variables are created in MATLAB they can be used in a wide variety of mathematical operations. In Chapter 1 the variables that were used in mathematical operations were all defined as scalars. This means that they were all 1×1 arrays (arrays with one row and one column that have only one element) and the mathematical operations were done with single numbers. Arrays, however, can be one-dimensional (arrays with one row, or with one column), two-dimensional (arrays with rows and columns), and even of higher dimensions. In these cases the mathematical operations are more complex. MATLAB, as its name indicates, is designed to carry out advanced array operations that have many applications in science and engineering. This chapter presents the basic, most common mathematical operations that MATLAB performs using arrays.

Addition and subtraction are relatively simple operations and are covered first in Section 3.1. The other basic operations, multiplication, division and exponentiation, can be done in MATLAB in two different ways. One way, which uses the standard symbols (*, /, and ^), follows the rules of linear algebra and is presented in Sections 3.2 and 3.3. The second way, which is called element-by-element operations, is covered in Section 3.4. These operations use the symbols .*, ./, and .^ (a period is typed in front of the standard operation symbol). In addition, in both types of calculations, MATLAB has left division operators (.\ or \ ), which are also explained in Sections 3.3 and 3.4.

A note to first time users of MATLAB:

Although matrix operations are presented first and element-by-element operations next, the order can be reversed since the two are independent of each other. It is expected that almost every user has some knowledge of matrix operations and linear algebra, and thus will be able to follow the material covered in Sections 3.2 and 3.3 without any difficulty. Some readers, however, might prefer to read Section 3.4 first. MATLAB can be used with element-by-element operations in numerous applications that do not require linear algebra multiplication (or division) operations.