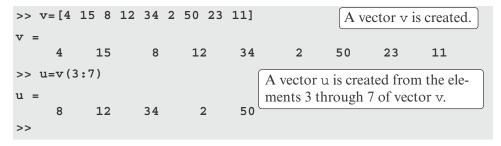## 2.6 USING A COLON : IN ADDRESSING ARRAYS

A colon can be used to address a range of elements in a vector or a matrix.

For a vector:

va(:)   Refers to all the elements of the vector *va* (either a row or a column vector).

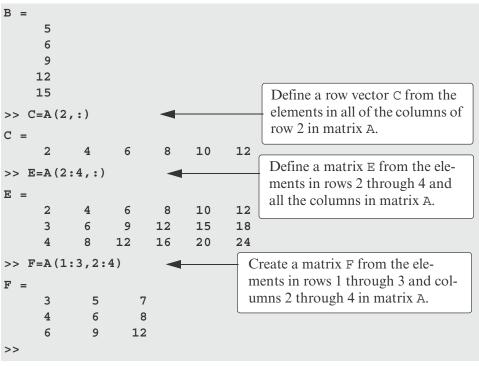*va*(*m:n*)   Refers to elements *m* through *n* of the vector *va*.
Example:

```
>> v=[4 15 8 12 34 2 50 23 11]                          A vector v is created.

v =
      4     15      8     12     34      2     50     23     11
>> u=v(3:7)                                    A vector u is created from the ele-
                                               ments 3 through 7 of vector v.
u =
      8     12     34      2     50
>>
```

For a matrix:

A(:,*n*)   Refers to the elements in all the rows of column *n* of the matrix *A*.

A(*n*,:)   Refers to the elements in all the columns of row *n* of the matrix *A*.

A(:,*m:n*)   Refers to the elements in all the rows between columns *m* and *n* of the matrix *A*.

A(*m:n*,:)   Refers to the elements in all the columns between rows *m* and *n* of the matrix *A*.

A(*m:n,p:q*)   Refers to the elements in rows *m* through *n* and columns *p* through *q* of the matrix *A*.

The use of the colon symbol in addressing elements of matrices is demonstrated in Tutorial 2-3.

Tutorial 2-3: Using a colon in addressing arrays.

```
>> A=[1 3 5 7 9 11; 2 4 6 8 10 12; 3 6 9 12 15 18; 4 8 12 16
20 24; 5 10 15 20 25 30]

                                              Define a matrix A with
                                              5 rows and 6 columns.
A =
      1      3      5      7      9     11
      2      4      6      8     10     12
      3      6      9     12     15     18        Define a column
      4      8     12     16     20     24        vector B from the
      5     10     15     20     25     30        elements in all of
                                                  the rows of column
>> B=A(:,3)                                       3 in matrix A.
```

Tutorial 2-3: Using a colon in addressing arrays. (Continued)

```
B =
     5
     6
     9
    12
    15
>> C=A(2,:)

C =
     2     4     6     8    10    12

>> E=A(2:4,:)

E =
     2     4     6     8    10    12
     3     6     9    12    15    18
     4     8    12    16    20    24

>> F=A(1:3,2:4)

F =
     3     5     7
     4     6     8
     6     9    12

>>
```

Define a row vector C from the elements in all of the columns of row 2 in matrix A.

Define a matrix E from the elements in rows 2 through 4 and all the columns in matrix A.

Create a matrix F from the elements in rows 1 through 3 and columns 2 through 4 in matrix A.

   In Tutorial 2-3 new vectors and matrices are created from existing ones by using a range of elements, or a range of rows and columns (using :). It is possible, however, to select only specific elements, or specific rows and columns of existing variables to create new variables. This is done by typing the selected elements or rows or columns inside brackets, as shown below:

```
>> v=4:3:34

v =
     4     7    10    13    16    19    22    25    28    31    34

>> u=v([3,   5,   7:10])

u =
    10    16    22    25    28    31

>> A=[10:-1:4; ones(1,7); 2:2:14; zeros(1,7)]

A =
    10     9     8     7     6     5     4
     1     1     1     1     1     1     1
     2     4     6     8    10    12    14
     0     0     0     0     0     0     0

>> B = A([1,3],[1,3,5:7])
```

Create a vector v with 11 elements.

Create a vector u from the 3rd, the 5th, and the 7th through 10th elements of v.

Create a 4×7 matrix A.

Create a matrix B from the 1st and 3rd rows, and 1st, 3rd, and the 5th through 7th columns of A.