generates (creates) the x matrix as

```
x =
     9    17    25
    10    18    26
    11    19    27
```

To terminate `for` and `while`, the `break` statement is used.

3.6. Illustrative Examples

Example 3.6.1.

Find the values of a, b, and c as given by the following expressions

$$a = 5x^2 - 6y + 7z,\ b = \frac{3y^2}{4x - 5z^3} \text{ and } c = \left(1 + \frac{1}{x^2}\right)^{-1}$$

if $x = 10$, $y = -20$ and $z = 30$.

Solution.

In the MATLAB Command Window, to find a, we type the statements

```
>> x=10; y=-20; z=30; a=5*x^2-6*y+7*z
```

Pressing the Enter key, we have the result. In particular,

```
a =
   830
```

That is, $a = 830$.

To find b, we type

```
>> x=10; y=-20; z=30; b=3*y^2/(4*x-5*z^3)
```

and the value for b is found. In particular,

```
b =
   -0.0089
```

Thus, $b = -0.0089$

Finally, to find the value of c, we have

```
>> x=10; y=-20; z=30; c=(1+1/x^2)^-1
```

and pressing the Enter key, the c value is displayed. That is, making use of

```
c =
    0.9901
```

we conclude that $c = 0.9901$. □

Example 3.6.2.

Use MATLAB to calculate the value of $e^{17.11}$.

Solution.

This problem is solved as follows:

```
» e=exp(17.11)
e =
  2.6964e+007
```

□

Example 3.6.3.

Given the complex number $N = 13 - 7i$. Using MATLAB, perform the following numerical calculations:

a. Find the magnitude of N.

b. Find the phase angle of N.

c. Determine the complex conjugate of N.

Solution.

The complex number is downloaded as

```
>> N=13-7i
```

We can use either `i` or `j` for the imaginary number. The three problems can be straightforwardly solved. In particular,

a.

```
>> abs(N)
ans =
   14.7648
```

b.

```
>> angle(N)
ans =
   -0.4939
```

c.

```
>> conj(N)
ans =
  13.0000 + 7.0000i
```

□

Example 3.6.4.

For a shell with an external diameter $r_1 = 10$ and internal diameter $r_2 = 2$, find the volume which is given by the following formula:

$$V = \frac{4}{3}\pi\left(r_1^3 - r_2^3\right).$$

Solution.

To solve this problem, the MATLAB statement (typed in the Command Window) is

```
>> r1=10; r2=2; V=(4*pi/3)*(r1^3-r2^3)
```

Here, we use the `pi` which is the constant π. Then, using the numerical result displayed,

```
V =
  4.1553e+003
```

we conclude that $V = 4155$. □

Example 3.6.5.

For a shell with an external diameter $r_1 = 3, 4, 5, 6, 7, 8, 9$, and 10 and internal diameter $r_2 = 2$, find the values for volume $V = \frac{4}{3}\pi\left(r_1^3 - r_2^3\right)$. Calculate and plot a nonlinear function $V = f(r_1)$ for the r_1 given.

Solution.

In the MATLAB Command Window we type

```
>> r1=3:1:10; r2=2; V=(4*pi/3)*(r1.^3-r2^3); plot(V,r1)
```

Then, values for V are found to be

```
V =
  1.0e+003 *
     0.0796     0.2346     0.4901     0.8713     1.4032     2.1112     3.0201
4.1553
```

To plot the nonlinear function $V = f(r_1)$, we have

```
>> r1=3:1:10; r2=2; V=(4*pi/3)*(r1.^3-r2^3);  plot(V,r1)
```

and the plot is illustrated in Figure 3.13.

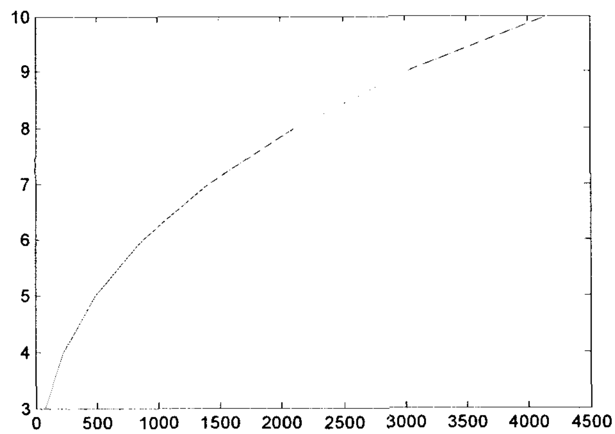

Figure 3.13. Nonlinear function: volume versus the external radius $V=f(r_1)$ □

Example 3.6.6.

Use the `linspace` function and increment method to create a vector A with 15 equally spaced values, beginning with 7.0 and ending with 47.5.

Solution.

Using `linspace`, in the Command Window, we type

```
>> A=linspace(7.0, 47.5, 15)
```

where 7.0 is the first (initial) value, 47.5 is the final value, and 15 is the number of values to be displayed. Pressing the Enter key, the following is displayed in the Command Window:

```
A =
  Columns 1 through 9
   7.0000    9.8929   12.7857   15.6786   18.5714   21.4643   24.3571   27.2500   30.1429
  Columns 10 through 15
  33.0357   35.9286   38.8214   41.7143   44.6071   47.5000
```

The first value is 7.0, the final value is 47.5, and 15 numbers are displayed.

The increment method can be used. The increment value is found using the equation

$$\text{Increment} = \frac{\text{Final Value} - \text{Initial Value}}{\text{Number of Increments} - 1}.$$

Thus, we enter the following statement,

```
>> Increment=(47.5-7.0)/(15-1)
```

and press the Enter key. We have

```
Increment =
    2.8929
```

Because a value of 2.8929 is now assigned to be the increment, we can enter

```
>> A=[7.0:Increment:47.5]
```

The result is

```
A =
Columns 1 through 9
7.0000    9.8929    12.7857    15.6786    18.5714    21.4643    24.3571    27.2500    30.1429
Columns 10 through 15
33.0357    35.9286    38.8214    41.7143    44.6071    47.5000
```

□

Example 3.6.7.

Use `linspace` and apply the increment method to create vector B with starting (initial) value of 7 and final (ending) value of 23 with increment of 0.16 between values. Display only the 18th value in each case.

Solution.

Increment method. We enter

```
>> B=[7:0.16:23];
>> Display=B(18)
```

Here, the first line is the vector, and the second line indicates the value we wish to display. Pressing the Enter key, we have

```
Display =
    9.7200
```

Using the `linspace` function, we must first find the number of values that will be found using the following equation:

$$\text{Number of Values} = \frac{\text{Final Value} - \text{Initial Value}}{\text{Increment}} + 1.$$

Thus, in the command line, enter the equation and press Enter key:

```
>> Number=((23-7)/(.16))+1
Number =
   101
```

This means that the 101 values would have been displayed under the given conditions. Now, using `linspace` and `display`, we type

```
>> B=linspace(7,23,Number);
>> Display=B(18)
Display =
    9.7200
```

□

Example 3.6.8.

For the given the matrices $C = \begin{bmatrix} 6 & 9 & 5 & 1 \\ 8 & 7 & 2 & 3 \\ 1 & 3 & 4 & 4 \\ 5 & 2 & 8 & 2 \end{bmatrix}$ and $D = \begin{bmatrix} 4 & 8 \\ 3 & 7 \\ 2 & 3 \\ 5 & 1 \end{bmatrix}$, in MATLAB perform the following:

a. Create matrix E1 with the two middle columns of C using the colon operator.
b. Create matrix E2 with rows 1 and 2 and columns 2 and 3 of C using the colon operator.
c. Create matrix E3 by placing E1 and D side by side.
d. Find the product of C_{24} and D_{12}.

Solution.

First, we download the matrices C and D as

```
>> C=[6 9 5 1;8 7 2 3;1 3 4 4;5 2 8 2];
>> D=[4 8;3 7;2 3;5 1];
```

a. The matrix E1 is created as

```
>> E1=C(:,2:3)
E1 =
     9     5
     7     2
     3     4
     2     8
```

b. The matrix E2 is generated as

```
>> E2=C(1:2,2:3)
E2 =
     9     5
     7     2
```

c. The matrix E3 is created as

```
>> E3=[E1,D]
E3 =
     9     5     4     8
     7     2     3     7
     3     4     2     3
     2     8     5     1
```

d. The product of the value in row 2 and column 4 of matrix C and the value in row 1 and column 2 in matrix D is found as

```
>> C(2,4)*D(1,2)
```

We use the numbers for row and column location in matrices C and D. Pressing the Enter key, we have

```
ans =
    24
```

In fact, the product of 8 and 3 is 24. □

Example 3.6.9.

We have the following arrays F = [3 21 6 17], G = [4 27 9 3], and H = [1 2 9 15].

a. Combine F, G and H into a matrix K1 such that F is in the first row of K1, G is in the second row of K1, and H is the third row of K1.

b. Combine F, G and H into a matrix K2 such that F is in the first column of K2, G is in the second column of K2, and H is the third column of K2.

Solution.

First, we download the arrays assigned as

```
>> F=[3 21 6 17]; G=[4 27 9 3]; H=[1 2 9 15];
```

a. The matrix K1 is created and displayed using the following statement

```
>> K1=[F;G;H]
K1 =
     3    21     6    17
     4    27     9     3
     1     2     9    15
```

b. The matrix K2 is generated as

```
>> K2=[F',G',H']
K2 =
     3     4     1
    21    27     2
     6     9     9
    17     3    15
```

Here, the transpose symbol ' transforms a horizontal array into a vertical one. □

Example 3.6.10.

Given matrices A and B as $A=\begin{bmatrix}7 & 2\\ 3 & 1\end{bmatrix}$ and $B=\begin{bmatrix}2 & 3\\ -4 & -5\end{bmatrix}$, calculate the following:

a. A + B
b. A – B
c. 2*B
d. A/4
e. A.*B
f. B.*A
g. A*B
h. B*A
k. A.^2
l. A^2
m. A.^B
n. A./B

using pencil and paper. Verify the results using MATLAB.

Solution.

First, we download matrices A and B as

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

If one wants to display these matrices, the first semicolon outside the brackets must be replaced by a comma, and remove the second semicolon. In particular, typing

```
>> A=[7 2;3 1], B=[2 3;-4 -5]
```

and pressing the Enter key, the following is displayed

```
A =
     7     2
     3     1
B =
     2     3
    -4    -5
```

a. A + B

Typing

```
>> A+B
```

and pressing the Enter key, we have

```
ans =
  9  5
 -1 -4
```

b. A – B

Making use of

```
>> A-B
```

and pressing the Enter key, one has

```
ans =
     5    -1
     7     6
```

c. 2*B

It is obvious that the following must be typed:

```
>> 2*B
```

The answer is

```
ans =
     4     6
    -8   -10
```

d. A/4

To find the resulting matrix, we have

```
>> A/4
```

Then, pressing the Enter key, we find

```
ans =
     1.7500    0.5000
     0.7500    0.2500
```

e. A.*B

To multiply the matrices' elements (entries), we type

```
>> A.*B
```

The answer is displayed pressing the Enter key. In particular,

```
ans =
    14     6
   -12    -5
```

f. B.*A

We type

```
>> B.*A
```

and have the resulting answer:

```
ans =
    14     6
   -12    -5
```

g. A*B

To multiply the matrices, we type

```
>> A*B
```

By pressing the Enter key, the resulting matrix is found to be

```
ans =
      6    11
      2     4
```

h. B*A

The matrices are multiplied using the following statement:

```
>> B*A
```

We have the following answer pressing the Enter key:

```
ans =
  23   7
 -43  -13
```

k. A.^2

In order to squire the elements (entries) of matrix A, we type in the Command Window

```
>> A.^2
```

Pressing the Enter key, we have

```
ans =
     49     4
      9     1
```

l. A^2

Here we need to multiply matrix A by A. We have

```
>> A^2
```

The result is

```
ans =
     55    16
     24     7
```

m. A.^B

We type

```
>> A.^B
```
and the following answer is displayed:
```
ans =
   49.0000    8.0000
    0.0123    1.0000
```
n. A./B

By typing in the Command Window
```
>> A./B
```
we obtain
```
ans =
    3.5000    0.6667
   -0.7500   -0.2000
```
□

Example 3.6.11.

Given the matrix $A = \begin{bmatrix} 12.11 & -7.9 & 9.23 \\ 5.06 & 6.35 & 21.7 \\ -3.34 & 2.67 & 14.38 \end{bmatrix}$, using MATLAB do the following:

a. Find the natural logarithm of the absolute value of each element of A.
b. Find the base 10 logarithm of the absolute value of each element of A.
c. Find the square root of each element in A.
d. Calculate the hyperbolic cosine of each entry of A.
e. Round each element in A to the nearest integer.
f. Round each element in A to the next higher integer.
g. Truncate each element of A to the next lower integer toward zero.
h. Find the sum of the elements in each column of A.
k. Find the product of the elements in each row of A.
l. Find the maximum value in each row of A.
m. Find the minimum value in each row of A.
n. Sort the elements in each column of A in ascending order.
o. Sort the elements in each row of A in ascending order.
p. Find the mean of the values in each column of A.
q. Find the size of A.

Solution.

First, we download matrix A as
```
>> A=[12 -7 9;5 6 21;-3 2 14];
```
To solve the problems and find the numerical values, we type in the Command Window using the corresponding statements listed below. To find the answers, the Enter key must be pressed. We have

Part a
```
>> log(abs(A))
ans =
    2.4940    2.0669    2.2225
    1.6214    1.8485    3.0773
    1.2060    0.9821    2.6658
```
Part b
```
>> log10(abs(A))
ans =
    1.0831    0.8976    0.9652
    0.7042    0.8028    1.3365
    0.5237    0.4265    1.1578
```

Part c

```
>> sqrt(A)
ans =
   3.4799          0 + 2.8107i   3.0381
   2.2494          2.5199        4.6583
   0 + 1.8276i     1.6340        3.7921
```

Part d

```
>> cosh(A)
ans =
  1.0e+009 *
    0.0001    0.0000    0.0000
    0.0000    0.0000    1.3279
    0.0000    0.0000    0.0009
```

It is evident that another format should be used. For example, using the `format short e` we find

```
>> cosh(A)
ans =
  8.1377e+004  5.4832e+002  4.0515e+003
  7.4210e+001  2.0172e+002  6.5941e+008
  1.0068e+001  3.7622e+000  6.0130e+005
```

Part e

```
>> round(A)
ans =
    12    -8     9
     5     6    22
    -3     3    14
```

Part f

```
>> ceil(A)
ans =
    13    -7    10
     6     7    22
    -3     3    15
```

Part g

```
>> fix(A)
ans =
    12    -7     9
     5     6    21
    -3     2    14
```

Part h

```
>> sum(A)
ans =
   13.8300    1.1200   45.3100
```

Part k

```
>> prod(A')'
ans =
 -883.0249
  697.2427
 -128.2380
```

Part l

```
>> max(A)
ans =
   12.1100    6.3500   21.7000
```

Part m

```
>> min(A')'
ans =
   -7.9000
    5.0600
   -3.3400
```

Part n

```
>> sort(A)
ans =
   -3.3400   -7.9000    9.2300
    5.0600    2.6700   14.3800
   12.1100    6.3500   21.7000
```

Part o

```
>> sort(A')'
ans =
   -7.9000    9.2300   12.1100
    5.0600    6.3500   21.7000
   -3.3400    2.6700   14.3800
```

Part p

```
>> mean(A)
ans =
    4.6100    0.3733   15.1033
```

Part q

```
>> size(A)
ans =
     3     3
```

□

Example 3.6.12.

Given polynomials $f = 15x^3 - 7x^2 + 2x + 4$ and $g = 9x^2 - 17x + 3$, do the following problems:

a. find the product of f and g,

b. find the quotient and remainder of f divided by g,

c. find the roots of g.

Solution.

We download two polynomials as

```
>> f=[15 -7 2 4]; g=[9 -17 3];
```

a. The product of f and g is found using the `conv` function. In particular,

```
>> fg=conv(f,g)
fg =
   135  -318   182   -19   -62    12
```

b. Quotient and remainder of f divided by g is found using `deconv`. Specifically,

```
>> [Quotient, Remainder]=deconv(f,g)
Quotient =
    1.6667    2.3704
Remainder =
    0.0000         0   37.2963   -3.1111
```

c. The `roots` function is applied to find the roots of f.

```
>> roots(f)
ans =
   0.4672 + 0.5933i
   0.4672 - 0.5933i
  -0.4676
```

□

Example 3.6.13.

Write an m-file which will generate a table of conversions from inches to centimeters using the conversion factor 1 inch = 2.54 cm. Prompt the user to enter the starting number of inches. Increment the inch value by 3 on each line. Display a total of 10 lines. Include a title and column heading in the table.

Solution.

The m-file should be written. Fiurthermore, to execute an m-file, MATLAB must be able to find it. This means that a directory in MATLAB's path must be found. The current working directory is always on the path. To display or change the path, we use the `path` function. To display or change the working directory, the user must use `cd`. As usual, `help` will provide more information.

To solve the problem, the following m-file is written. Comments are identified by the `%` symbol.

```
%This program will generate a table of conversions from inches to centimeters.
%The user will be allowed to enter the initial inch value.
%The values will be incremented by 3 inches, with a %total of 10 lines printed.
%User input is equal to the variable inch_initial
inch_initial=input('Enter the initial length in inches:  ')
%Calculate largest inch value to be printed (limit table to 10 lines of text)
inch_final=inch_initial+5*9;
%Increment value by 3
inches=[inch_initial:3:inch_final];
%Convert inches to centimeters
centimeters=inches.*2.54;
%Table of values
table=[inches; centimeters];
%Format the table
fprintf('Conversion from Inches to Centimeters\n')
fprintf('Inches     Centimeters\n\n')
fprintf('%6.3f      %6.3f\n', table)
```

Assigning the initial length to be 10, the following results are displayed:

```
Enter the initial length in inches:  10

inch_initial =
    10

Conversion from Inches to Centimeters
Inches Centimeters

10.000 25.400
13.000 33.020
16.000 40.640
19.000 48.260
22.000 55.880
25.000 63.500
28.000 71.120
31.000 78.740
34.000 86.360
37.000 93.980
40.000 101.600
43.000 109.220
46.000 116.840
49.000 124.460
52.000 132.080
55.000 139.700
```

□

Example 3.6.14.

Write an m-file that will calculate the area of circles ($A = \pi r^2$) with radii ranging from 3 to 8 meters at an increment between values entered by the user in the Command Window. Generate the results in a table using `disp` and `fprintf`, with radii in the first column and areas in the second column. When `fprintf` is used, print the radii with two digits after the decimal point and the areas with four digits after the decimal point.

Solution.

To solve the problem, the MATLAB script is developed and listed below.

```
%This program calculates area of circles with radii ranging from 3 to 8
%An increment is specified by the user
%One table is displayed using disp and the other using fprintf
%Radii have two digits and areas have four digits after the decimal point
%User input is equal to the variable increment
increment=input('Enter the increment value for the radii:  ')
%Increment radii
r=[3:increment:8];
%Equation for area
A=pi*r.^2;
table1=[r; A];
%Transpose table1
table2=table1';
%Disp is used to generate the table
disp('Areas of Circles with Different Radii')
disp('  Radius      Area')
disp('  Meters      Square Meters')
%Transposed table printed
disp(table2)
%fpintf is used to generate the table
fprintf('\n')
fprintf('Areas of Circles with Different Radii\n')
fprintf('Radius         Area\n')
fprintf('Meters         Square Meters\n')
fprintf('%5.2f      %10.4f\n', table1)
%Table automatically transposed
```

Assigning the increment to be 1, we have

```
Enter the increment value for the radii:  1

increment =
     1

Areas of Circles with Different Radii
  Radius      Area
  Meters      Square Meters
    3.0000   28.2743
    4.0000   50.2655
    5.0000   78.5398
    6.0000  113.0973
    7.0000  153.9380
    8.0000  201.0619

Areas of Circles with Different Radii
Radius   Area
Meters   Square Meters
 3.00     28.2743
 4.00     50.2655
 5.00     78.5398
 6.00    113.0973
 7.00    153.9380
 8.00    201.0619
```

Example 3.6.15.

Write an m-file which allows the user to enter (download) the temperatures in degrees Fahrenheit and return the temperature in degrees Kelvin. Use the formulas C° = 5(F° – 32)/9 and K = C° + 273.15. The output should include both the Fahrenheit and Kelvin temperatures. Make three variations of the output as:

a. Output temperatures as decimals with 5 digits following the decimal point,

b. Output temperatures in exponential format with 7 significant digits,

c. Output temperatures with 4 significant digits.

Solution

The following MATLAB script allows us to solve the problem:

```
%This program allows the user to enter a temperature in degrees Fahrenheit and will
%return the temperature in degrees Kelvin in three different formats.
%User input is equal to the variable "Fahrenheit"
Fahrenheit=input('Enter a temperature in degrees Fahrenheit:  ')
%Convert from Fahrenheit to Kelvin
Kelvin=5*(Fahrenheit-32)/9+273.15;
fprintf('Kelvin:\n')
%Part a: output temperature as decimals with 5 digits following the decimal point
fprintf('%.5f  \n', Kelvin)
%Part b:  output temperature in exponential format with 7 significant digits
fprintf('%.6e  \n', Kelvin)
%Part c:  output temperature in general format with 4 significant digits
fprintf('%.4g  \n\n', Kelvin)
```

For 100°F, the results displayed are given below:

```
Enter a temperature in degrees Fahrenheit:  100

Fahrenheit =
   100

Kelvin:
310.92778
3.109278e+002
310.9
```

□

Example 3.6.16.

A spring's potential energy is found as $E = kx^2/2$, where k is the spring constant; x is the spring displacement. The spring force is $F = kx$.

Using the data for five different springs as given in Table 3.2, write an m-file to find the displacement and potential energy stored in each spring. Output the results in a table that displays the spring number, displacement in meters, and potential energy in joules. The calculated values should have three digits following the decimal point.

Table 3.2. Spring Data

Spring	1	2	3	4	5
Force (N)	23	123	5	79	8
Spring constant, k	145	33	12	17	34

Solution.

The problem is solved by making use of the following MATLAB script:

```
%This program calculates the displacement and potential energy of springs.
%The force and spring constant will be entered by the user for each of five springs
%The program will print the output in three columns
%Input the vector values for force and constant "k"
force=[23 123 5 79 8];
k=[145 33 12 17 34];
%Increment spring number
spring_number=[1:1:5];
```

```
%Calculate displacement
displacement=force./k;
potential=(k.*(displacement.^2))/2;
%Calculate potential energy stored in the spring
%Table of values
table=[spring_number;displacement;potential];
fprintf('Spring Number     Displacement Potential Energy\n')
fprintf('                  (meters)            (Joules)\n')
%print output
fprintf('%1.0f              %7.3f              %7.3f\n', table)
```

The results displayed in the Command Window are documented below:

```
Spring Number         Displacement          Potential Energy
                      (meters)              (Joules)
1                     0.159                 1.824
2                     3.727                 229.227
3                     0.417                 1.042
4                     4.647                 183.559
5                     0.235                 0.941
```

□

Example 3.6.17.

The formula for the volume of a truncated cone is $V=\dfrac{\pi R^3}{3\tan\theta}-\dfrac{\pi}{3}(R-y\tan\theta)^2\left(\dfrac{R}{\tan\theta}-y\right)$.

Here, y is the height; R is the radius of the base; θ is the angle in radians formed by the centerline and the side of the cone at the ape.

Find the volume for radii $R = 1$, 2 and 3 meters if $y = 5$ meters and $\theta = 20°$. Calculate three volumes.

Solution.

The m-file is written. In particular,

```
%This program calculates volume of a truncated cone after the user inputs cone height,
%radius of the base, and the angle in radians formed by the centerline and the side of
%the cone at the apex
%Prompt user to input data, input assigned to the specified variable names
y=input('Enter the height of the cone in meters:  \n')
R=input('Enter the three radii in meters:  \n')
t=input('Enter the angle in degrees:  \n')
%Convert angle from degrees to radians
t=t*pi/180;
%Calculate volumes for the tree radii
V=((pi*R.^3)/(3*tan(t)))-(pi/3)*((R-y*tan(t)).^2).*((R./(tan(t)))-y);
%Display calculated output for three radii
disp(V)
```

The results displayed in the Command Window are documented below:

```
Enter the height of the cone in meters:
3
y =
     3
Enter the three radii in meters:
[1 2 3]
R =
     1     2     3
Enter the angle in degrees:
20
t =
    20
    2.8794    20.8627    57.6956
```

The three volumes are found to be 2.8794, 20.8627, and 57.6956. □

Example 3.6.18.

Write a MATLAB script which accepts the radius and height as inputs and returns the volume of the cone with those dimensions.

Solution.

The script (ch3618.m) is given below.

```
function V=ch3618(r,h)
%This program accepts the radius and height of a cone and calculates the cone volume
%User types in the function's name the variables separated by a comma
V=(1/3)*(pi*r.^2)*h;
```

The numerical results for $r = 5$ and $h = 10$ are

```
>> ch3618(5,10)

ans =
  261.7994
```

Thus, $V = 261.7994$. □

Example 3.6.19.

Write an m-file that computes the time t at which an object thrown vertically upward with the initial velocity v will reach a height h. There are two solutions for t in the height equation $h(t) = vt - \frac{1}{2}gt^2$ because the equation is quadratic. Test the file if $h = 100$ m, $v = 50$ m/sec, and $g = 9.81$ m/sec^2.

Solution.

The MATLAB script ch3619.m is

```
function t=ch3619(g,v,h)
%This program computes the time at which an object will reach a certain height.
%The user must input the gravity constant, the initial velocity and the height
%Two outputs will be displayed for t
%The first t is the time the object reaches the specific height on the way up
%The second t is the time when the object reaches/passes the height on the way down
%Terms of the equation are separated in ascending order of the exponents
C=[-.5*g;v;-h]
%Function roots is used to compute the two values of time
T=roots(C)
```

Numerical results are

```
>> ch3619(9.81,50,100)

C =
   -4.9050
   50.0000
 -100.0000

T =
    7.4612
    2.7324
```

□

Example 3.6.20.

Write the MATLAB file to solve linear algebraic equations. Develop an m-file in order to solve the following sets of linear algebraic equations:

a. $6x - 3y + 4z = 41$
$12x + 5y - 7z = -26$
$-5x + 2y + 6z = 14$

b. $12x - 5y = 11$
$-3x + 4y + 7z = -3$
$6x + 2y + 3z = 22$

c. $2.5x_4 + 5x_3 + x_1 - 2x_2 = -4$
$25x_2 - 6.2x_3 + 18x_4 + 10x_1 = 2.9$
$28x_4 + 25x_1 - 30x_2 - 15x_3 = -5.2$
$-3.2x_1 + 12x_3 - 8x_4 = -4.$

Solution.

The following m-file is written:

```
%This program solves linear equations using the downloaded matrices
%Three sets of matrices are
Aa=[6 -3 4;12 5 -7;-5 2 6]; Ba=[41;-26;14];
Ab=[12 -5 0;-3 4 7;6 2 3]; Bb=[11;-3;22];
Ac=[1 -2 5 2.5;10 25 -6.2 18;25 -30 -15 28;-3.2 0 12 -8]; Bc=[-4;2.9;-5.2;-4];
%Solve the equations
part_a=inv(Aa)*Ba;
part_b=(Ab^(-1))*Bb;
part_c=Ac\Bc;
%Print the results
fprintf('Part a: \n')
fprintf(' x       y         z \n')
fprintf('%3.2f       %3.2f  %3.2f \n\n', part_a)
fprintf('Part b: \n')
fprintf(' x         y           z \n')
fprintf('%3.2f      %3.2f  %3.2f \n\n', part_b)
fprintf('Part c: \n')
fprintf(' x1        x2     x3      x4 \n')
fprintf('%3.2f      %3.2f  %3.2f  %3.2f \n\n', part_c)
```

The displayed results are

```
Part a:
x      y      z
2.00   -3.00  5.00

Part b:
x      y      z
3.00   5.00   -2.00

Part c:
x1     x2     x3     x4
0.13   0.20   -0.55  -0.38
```

Thus, the solutions of the algebraic equations are found.

□

Example 3.6.21.

Electric circuits are described (modeled) using Kirchhoff's voltage and current laws. The electric circuit under consideration is described by the following set of five algebraic equations:

$R_1 i_1 + R_2 i_2 - v_1 = 0$
$-R_2 i_2 + R_3 i_3 + R_5 i_5 = 0$
$v_2 + R_4 i_4 - R_3 i_3 = 0$
$-i_1 + i_2 + i_3 + i_4 = 0$
$-i_4 - i_3 + i_5 = 0$

a. Calculate the five unknown currents (i_1, i_2, i_3, i_4, and i_5) using the following resistances and voltages as: $R_1 = 470$ ohm, $R_2 = 300$ ohm, $R_3 = 560$ ohm, $R_4 = 100$ ohm, $R_5 = 1000$ ohm, $v_1 = 5$V, and $v_2 = 10$V. Label the answers with current number and units. .

b. Using the resistances given above and $v_1 = 5$V, find the range of positive voltages v_2 for which none of the currents exceeds 50 mA. The currents may be positive or negative. None of the currents may be less than – 50 mA or greater than 50 mA.

Solution.

The MATLAB script is documented below.

```
%This program calculates the value of five currents using Kirchhoff's voltage law
%Part A. Variable values
R1=470; R2=330; R3=560; R4=100; R5=1000; v1=5; v2=10;
%Matrices used to solve the equation
A=[R1 R2 0 0 0;0 -R2 R3 0 R5;0 0 -R3 R4 0; -1 1 1 1 0;0 0 -1 -1 1];
B=[v1;0;-v2;0;0];
%Vector for current number
I=[1:5];
%Solve the linear algebraic equation
C=(A\B)';
%Create a table to output current number and current value
table=[I;C];
%Print a table that outputs the current values
fprintf('Part A: \n')
fprintf('Current Number              Current Value \n')
fprintf('                          (Amperes)     \n')
fprintf('--------------            ------------- \n')
fprintf('  %4.0f                     %8.6f\n', table)
%Part B:
%Form a loop to find the range of positive voltages v2 (currents cannot exceed 0.05A)
%value of subscript n
n=1;
%Use a range of v2 between 0V and 30V, incrementing by 0.1
%The absolute value of current (vector C) is used to avoid 2 loops
%Subscript n is incriminated by 1
for v2=0:0.1:30
   B=[v1;0;-v2;0;0];
   C=A\B;
   if abs(C)<0.05
      v(n)=v2;
       n=n+1;
   end
end
%Find minimum and maximum values of v2 and assign to two variables
v_max=max(v);
v_min=min(v);
%Print the output
fprintf('\n\nPart B: \n')
fprintf('The range of voltages v2 for which no currents are bellow\n')
fprintf('    -50mA or above 50mA is %4.1f V to %4.1f V. \n\n', v_min, v_max)
```

The results are

```
Part A:
Current Number        Current Value
                      (Amperes)
--------------        -------------
      1                0.004178
      2                0.009201
      3                0.014391
      4               -0.019413
      5               -0.005022

Part B:
The range of voltages v2 for which no currents are bellow
       -50mA or above 50mA is  0.0 V to 24.7 V.
```

□

Example 3.6.22.

The height, horizontal distance, and speed of a projectile launched with a speed v at an angle A to the horizontal line are given by the following formulas:

$$h(t) = vt\sin A - \tfrac{1}{2}gt^2,\ x(t) = vt\cos A \text{ and } v(t) = \sqrt{v^2 - 2vgt\sin A + g^2t^2}\ .$$

The projectile will strike the ground when $h(t) = 0$, and the time of the hit is $t_{hit} = 2\frac{v}{g}\sin A$.

Suppose that $A = 30°$, $v = 40$ m/s, and $g = 9.81$ m/s^2. Use logical operators to find the times (with the accuracy to the nearest hundredth of a second) when

a. The height is no less than 15 meters,

b. The height is no less than 15 meters and the speed is no greater than 36 m/sec.

Solution.

The following MATLAB script is developed to solve the problem.

```
%This program finds the times at which the conditions specified bellow are satisfied
%Parameters
A=30; vo=40; g=9.81;
%Part a:  Determine the times at which the height is no less than 15 meters.
%Increment time by 0.01
time=[0:0.01:5];
%Find height (h) for different values of time
h=vo*time.*sin(A*pi/180)-0.5*g*time.^2;
%Determine times when height is greater than 15 meters
times=find(h>=15);
%Find the maximum and minimum value for times
max_time=max(times)/100;
min_time=min(times)/100;
%Print the results
fprintf('\nThe height is at least 15 m when the time is between\n')
fprintf('     %.2f seconds and %.2f seconds.\n\n', min_time, max_time)
%Part b: determine t at which h is no less than 15 m and v is no more than 36 m/sec
%Increment time by 0.01
time=[0:0.01:5];
%Find height (h) and velocity (v) for different values of time
h=vo*time.*sin(A*pi/180)-0.5*g*time.^2;
v=sqrt((vo^2)-(2*vo*g*time.*sin(A*pi/180))+(g^2*time.^2));
%Determine times when heigh is greater than 15 m and velocity is no more than 36 m/sec
vector=find(h>=15 & v<=36);
%Find the maximum and minimum value for times
max_time=max(times);
min_time=min(times);
%Print the results
fprintf('\n\nThe height is at least 15 m and the velocity is no less than 36 m/s\n')
fprintf('when the time is between %f seconds and %f seconds.\n\n', min_time, max_time)
```