
6.2 **CONDITIONAL STATEMENTS**

A conditional statement is a command that allows MATLAB to make a decision of whether to execute a group of commands that follow the conditional statement, or to skip these commands. In a conditional statement, a conditional expression is stated. If the expression is true, a group of commands that follow the statement are executed. If the expression is false, the computer skips the group. The basic form of a conditional statement is:

`if` conditional expression consisting of relational and/or logical operators.

Examples:

```
if a < b
if c >= 5
if a == b
if a ~= 0
if (d<h) & (x>7)
if (x~=13) | (y<0)
```

All the variables must have assigned values.

- Conditional statements can be a part of a program written in a script file or a user-defined function (Chapter 7).
- As shown below, for every `if` statement there is an end statement.

The `if` statement is commonly used in three structures, `if-end`, `if-else-end`, and `if-elseif-else-end`, which are described next.

6.2.1 **The if-end Structure**

The `if-end` conditional statement is shown schematically in Figure 6-1. The figure shows how the commands are typed in the program, and a flowchart that symbolically shows the flow, or the sequence, in which the commands are executed. As the program executes, it reaches the `if` statement. If the conditional expres-

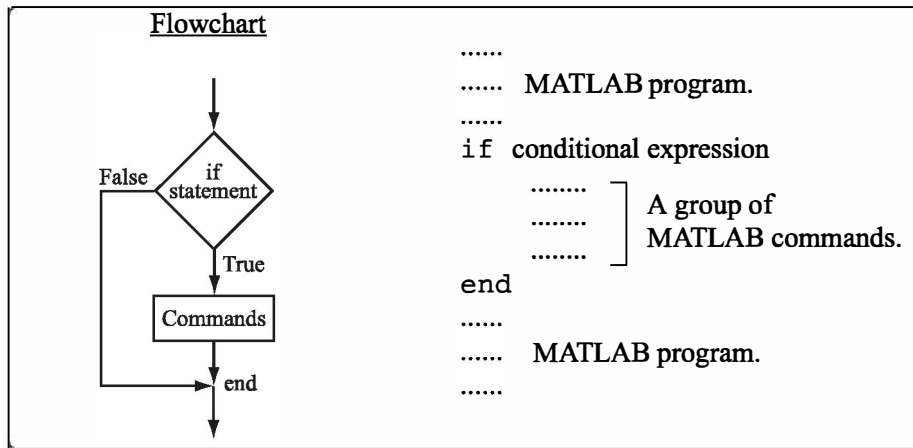


Figure 6-1: The structure of the if-end conditional statement.

sion in the `if` statement is `true` (1), the program continues to execute the commands that follow the `if` statement all the way down to the `end` statement. If the conditional expression is `false` (0), the program skips the group of commands between the `if` and the `end`, and continues with the commands that follow the `end`.

The words `if` and `end` appear on the screen in blue, and the commands between the `if` statement and the `end` statement are automatically indented (they don't have to be), which makes the program easier to read. An example where the `if-end` statement is used in a script file is shown in Sample Problem 6-2.

Sample Problem 6-2: Calculating worker's pay

A worker is paid according to his hourly wage up to 40 hours, and 50% more for overtime. Write a program in a script file that calculates the pay to a worker. The program asks the user to enter the number of hours and the hourly wage. The program then displays the pay.

Solution

The program in a script file is shown below. The program first calculates the pay by multiplying the number of hours by the hourly wage. Then an `if` statement checks whether the number of hours is greater than 40. If so, the next line is executed and the extra pay for the hours above 40 is added. If not, the program skips to the `end`.

```

t=input('Please enter the number of hours worked ');
h=input('Please enter the hourly wage in $ ');
Pay=t*h;
if t>40

```

```

    Pay=Pay+(t-40)*0.5*h;
end
fprintf('The worker's pay is $ %5.2f',Pay)

```

Application of the program (in the Command Window) for two cases is shown below (the file was saved as Workerpay):

```

>> Workerpay
Please enter the number of hours worked 35
Please enter the hourly wage in $ 8
The worker's pay is $ 280.00
>> Workerpay
Please enter the number of hours worked 50
Please enter the hourly wage in $ 10
The worker's pay is $ 550.00

```

6.2.2 The if-else-end Structure

The if-else-end structure provides a means for choosing one group of commands, out of a possible two groups, for execution. The if-else-end structure is shown in Figure 6-2. The figure shows how the commands are typed in the program, and includes a flowchart that illustrates the flow, or the sequence, in

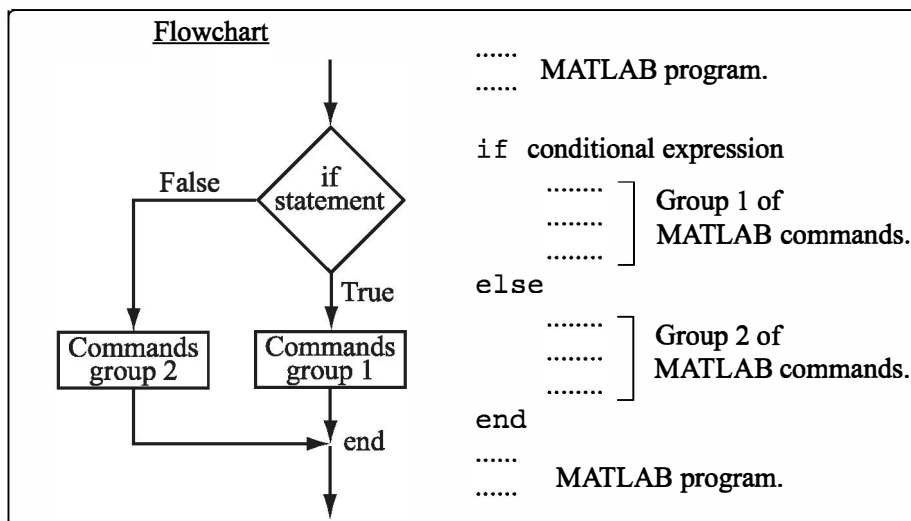


Figure 6-2: The structure of the if-else-end conditional statement.

which the commands are executed. The first line is an `if` statement with a conditional expression. If the conditional expression is true, the program executes group 1 of commands between the `if` and the `else` statements and then skips to the `end`. If the conditional expression is false, the program skips to the `elseif` and then executes group 2 of commands between the `elseif` and the `end`.

6.2.3 The `if-elseif-else-end` Structure

The `if-elseif-else-end` structure is shown in Figure 6-3. The figure shows how the commands are typed in the program, and gives a flowchart that illustrates the flow, or the sequence, in which the commands are executed. This structure includes two conditional statements (`if` and `elseif`) that make it possible to select one out of three groups of commands for execution. The first line is an `if` statement with a conditional expression. If the conditional expression is true, the program executes group 1 of commands between the `if` and the

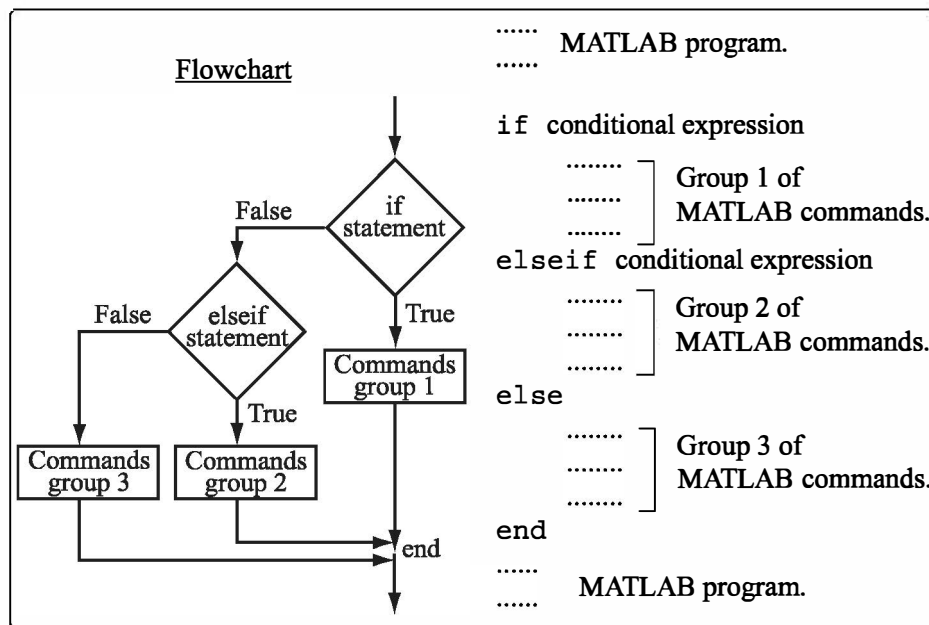


Figure 6-3: The structure of the `if-elseif-else-end` conditional statement.

`elseif` statements and then skips to the `end`. If the conditional expression in the `if` statement is false, the program skips to the `elseif` statement. If the conditional expression in the `elseif` statement is true, the program executes group 2 of commands between the `elseif` and the `else` and then skips to the `end`. If the conditional expression in the `elseif` statement is false, the program skips to the `else` and executes group 3 of commands between the `else` and the `end`.

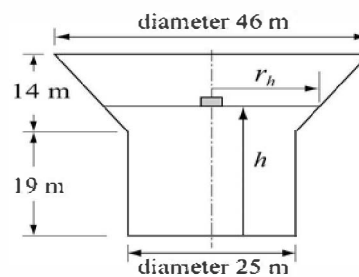
It should be pointed out here that several `elseif` statements and associ-

ated groups of commands can be added. In this way more conditions can be included. Also, the `else` statement is optional. This means that in the case of several `elseif` statements and no `else` statement, if any of the conditional statements is true the associated commands are executed; otherwise nothing is executed.

The following example uses the `if-elseif-else-end` structure in a program.

Sample Problem 6-3: Water level in water tower

The tank in a water tower has the geometry shown in the figure (the lower part is a cylinder and the upper part is an inverted frustum of a cone). Inside the tank there is a float that indicates the level of the water. Write a MATLAB program that determines the volume of the water in the tank from the position (height h) of the float. The program asks the user to enter a value of h in m, and as output displays the volume of the water in m^3 .



Solution

For $0 \leq h \leq 19$ m the volume of the water is given by the volume of a cylinder with height h : $V = \pi 12.5^2 h$.

For $19 < h \leq 33$ m the volume of the water is given by adding the volume of a cylinder with $h = 19$ m, and the volume of the water in the cone:

$$V = \pi 12.5^2 \cdot 19 + \frac{1}{3} \pi (h - 19) (12.5^2 + 12.5 \cdot r_h + r_h^2)$$

where $r_h = 12.5 + \frac{10.5}{14}(h - 19)$.

The program is:

```
% The program calculates the volume of the water in the
water tower.
h=input('Please enter the height of the float in meter ');
if h > 33
    disp('ERROR. The height cannot be larger than 33 m.')
elseif h < 0
    disp('ERROR. The height cannot be a negative number.')
elseif h <= 19
    v = pi*12.5^2*h;
    fprintf('The volume of the water is %7.3f cubic meter.\n',v)
```

```

else
    rh=12.5+10.5*(h-19)/14;
    v=pi*12.5^2*19+pi*(h-19)*(12.5^2+12.5*rh+rh^2)/3;
    fprintf('The volume of the water is %7.3f cubic meter.\n',v)
end

```

The following is the display in the Command Window when the program is used with three different values of water height.

```

Please enter the height of the float in meter 8
The volume of the water is 3926.991 cubic meter.

Please enter the height of the float in meter 25.7
The volume of the water is 14114.742 cubic meter.

Please enter the height of the float in meter 35
ERROR. The height cannot be larger than 33 m.

```

6.3 THE switch-case STATEMENT

The switch-case statement is another method that can be used to direct the flow of a program. It provides a means for choosing one group of commands for execution out of several possible groups. The structure of the statement is shown in Figure 6-4.

- The first line is the switch command, which has the form:

```
switch switch expression
```

The switch expression can be a scalar or a string. Usually it is a variable that has an assigned scalar or a string. It can also be, however, a mathematical expression that includes pre-assigned variables and can be evaluated.

- Following the switch command are one or several case commands. Each has a value (can be a scalar or a string) next to it (value1, value2, etc.) and an associated group of commands below it.
- After the last case command there is an optional otherwise command followed by a group of commands.
- The last line must be an end statement.

How does the switch-case statement work?

The value of the switch expression in the switch command is compared with the values that are next to each of the case statements. If a match is found, the group of commands that follow the case statement with the match are executed. (Only one group of commands—the one between the case that matches and either the

```

.....  MATLAB program.
.....

switch switch expression
  case value1
.....   ] Group 1 of commands.
.....   ]
  case value2
.....   ] Group 2 of commands.
.....   ]
  case value3
.....   ] Group 3 of commands.
.....   ]
  otherwise
.....   ] Group 4 of commands.
.....   ]
end
.....  MATLAB program.
.....

```

Figure 6-4: The structure of a switch-case statement.

case, otherwise, or end statement that is next—is executed).

- If there is more than one match, only the first matching case is executed.
- If no match is found and the otherwise statement (which is optional) is present, the group of commands between otherwise and end is executed.
- If no match is found and the otherwise statement is not present, none of the command groups is executed.
- A case statement can have more than one value. This is done by typing the values in the form: {value1, value2, value3, ...}. (This form, which is not covered in this book, is called a cell array.) The case is executed if at least one of the values matches the value of switch expression.

A Note: In MATLAB only the first matching case is executed. After the group of commands associated with the first matching case are executed, the program skips to the end statement. This is different from the C language, where break statements are required.

Sample Problem 6-4: Converting units of energy

Write a program in a script file that converts a quantity of energy (work) given in units of either joule, ft-lb, cal, or eV to the equivalent quantity in different units specified by the user. The program asks the user to enter the quantity of energy, its

current units, and the desired new units. The output is the quantity of energy in the new units.

The conversion factors are: $1 \text{ J} = 0.738 \text{ ft-lb} = 0.239 \text{ cal} = 6.24 \times 10^{18} \text{ eV}$.

Use the program to:

- Convert 150 J to ft-lb.
- Convert 2,800 cal to J.
- Convert 2.7 eV to cal.

Solution

The program includes two sets of switch-case statements and one if-else-end statement. The first switch-case statement is used to convert the input quantity from its initial units to units of joules. The second is used to convert the quantity from joules to the specified new units. The if-else-end statement is used to generate an error message if units are entered incorrectly.

```

Ein=input('Enter the value of the energy (work) to be converted: ');
EinUnits=input('Enter the current units (J, ft-lb, cal, or eV): ','s');
EoutUnits=input('Enter the new units (J, ft-lb, cal, or eV): ','s');
error=0;
switch EinUnits
case 'J'
    EJ=Ein;
case 'ft-lb'
    EJ=Ein/0.738;
case 'cal'
    EJ=Ein/0.239;
case 'eV'
    EJ=Ein/6.24e18;
otherwise
    error=1;
end
switch EoutUnits
case 'J'
    Eout=EJ;
case 'ft-lb'
    Eout=EJ*0.738;
case 'cal'
    Eout=EJ*0.239;
case 'eV'
    Eout=EJ*6.24e18;

```

Assign 0 to variable error.

First switch statement. Switch expression is a string with initial units.

Each of the four case statements has a value (string) that corresponds to one of the initial units, and a command that converts Ein to units of J. (Assign the value to EJ.)

Assign 1 to error if no match is found. Possible only if initial units were typed incorrectly.

Second switch statement. Switch expression is a string with new units.

Each of the four case statements has a value (string) that corresponds to one of the new units, and a command that converts EJ to the new units. (Assign the value to Eout.)


```
otherwise
    error=1;
end
if error
    disp('ERROR current or new units are typed incorrectly.')
else
    fprintf('E = %g %s',Eout,EoutUnits)
end
```

Assign 1 to error if no match is found. Possible only if new units were typed incorrectly.

If-else-end statement.

If error is true (nonzero), display an error message.

If error is false (zero), display converted energy.

As an example, the script file (saved as EnergyConversion) is used next in the Command Window to make the conversion in part (b) of the problem statement.

```
>> EnergyConversion
Enter the value of the energy (work) to be converted: 2800
Enter the current units (J, ft-lb, cal, or eV): cal
Enter the new units (J, ft-lb, cal, or eV): J
E = 11715.5 J
```