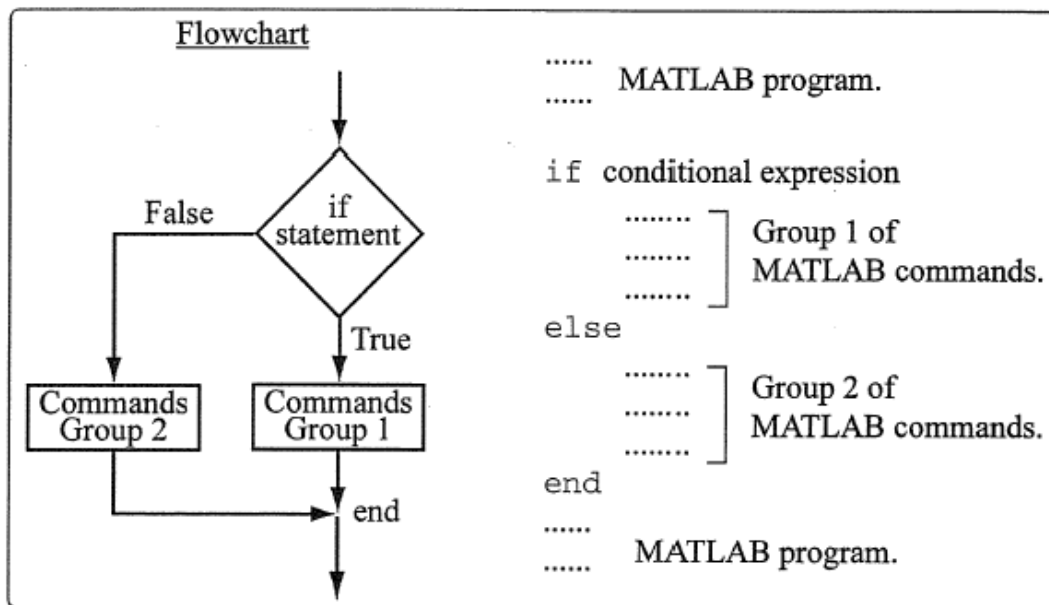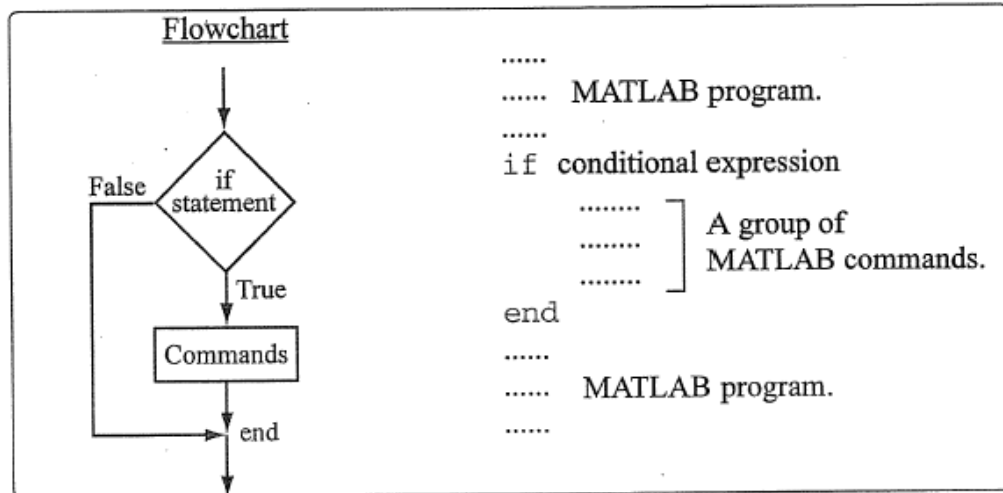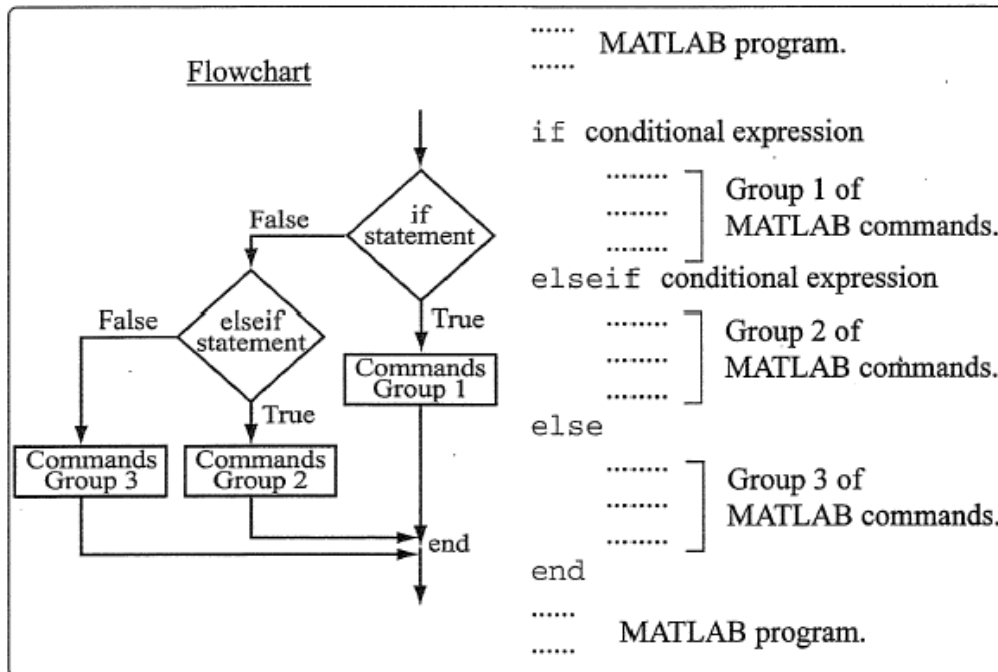# Flow Control in Matlab

If-end Statement

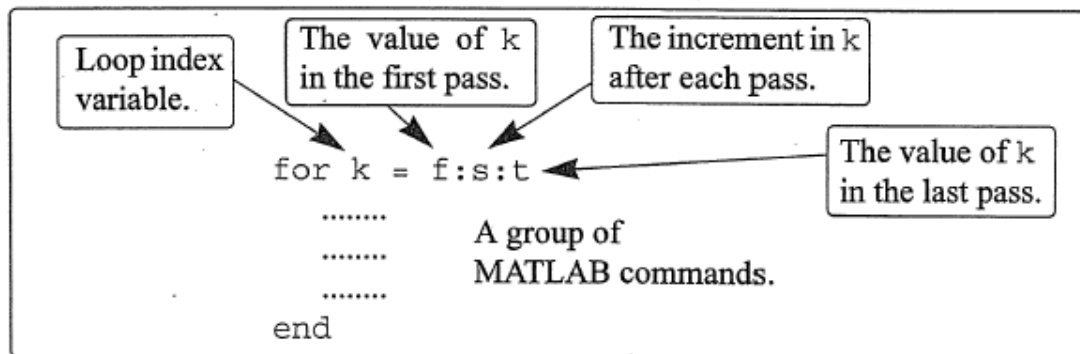

Figure 7-2: The structure of the if-else-end conditional statement.

Figure 7-3: The structure of the `if-elseif-else-end` **conditional statement.**
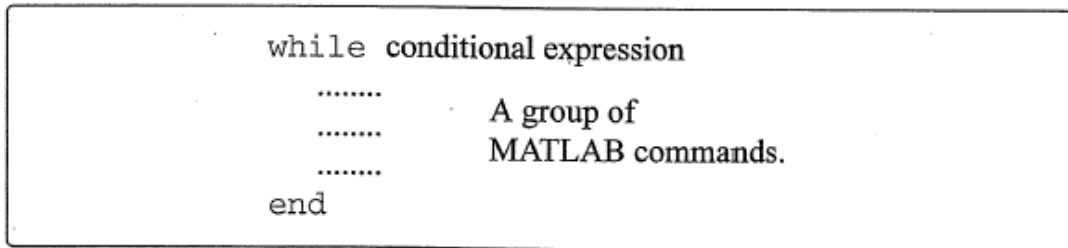
For-end-loops

Figure 7-5: The structure of a `for-end` **loop.**

Example:

```
for k= 1:3:10
  x = k^2
end
```

While-End loop

Figure 7-6: The structure of a while-end loop.

Example:

> **x=1;**
> **while** x<=15
> x=2*x
> **end**

Polynomial Evaluation

Matlab Functions: **Polyval** and **roots**

The polynomial $p(x) = 3x^2 + 2x + 1$ is evaluated at x = 5, 7, and 9 with

    p = [3 2 1];
    polyval (p,[5 7 9])

which results in

    ans =
        86   162   262

roots(p) results in:

ans =

 -0.3333 + 0.4714i
 -0.3333 - 0.4714i

To plot a polynomial
x=[-1.5:.1:6.7]
y=**polyval**(p,x)
plot(x,y)

Integration and Differentiation

Matlab Function   **quad** to calculate the integral of a function.

To compute the integral

$$\int_0^2 \frac{1}{x^3 - 2x - 5} dx$$

quad ('1./(x.^3-2*x-5', 0,2)

which results in:

ans =

   -0.4605

Solving Linear Equations in Matlab

Use matrix operations to solve the following system of linear equations.

$$4x - 2y + 6z = 8$$
$$2x + 8y + 2z = 4$$
$$6x + 10y + 3z = 0$$

Solution

Using the rules of linear algebra demonstrated earlier, the above system of equations can be written in the matrix form $AX = B$ or in the form $XC = D$:

$$\begin{bmatrix} 4 & -2 & 6 \\ 2 & 8 & 2 \\ 6 & 10 & 3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 8 \\ 4 \\ 0 \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} x & y & z \end{bmatrix} \begin{bmatrix} 4 & 2 & 6 \\ -2 & 8 & 10 \\ 6 & 2 & 3 \end{bmatrix} = \begin{bmatrix} 8 & 4 & 0 \end{bmatrix}$$

the solution of $AX = B$ is:

$$X = A^{-1}B$$

In MATLAB the last equation can be written by using the left division character:

$$X = A\backslash B$$