ENGR 1181 | MATLAB 08: For Loops 1

Preparation Material

Learning Objectives

1. Explain the iterative nature of loops
2. Use loops properly for repetitive processes, including the utilization of loop indices as counters and variables

Topics
Students will read Chapter 6.4.1 of the MATLAB book before coming to class. This preparation material is provided to supplement this reading.

Students will learn a basic understanding of a programming loop and how to properly code for-end loops in MATLAB. This material contains the following:

1. Key Definitions
2. Loop
3. for-end Loops
4. for-end Loops in MATLAB

1. Key Definitions

Loop – a command, or group of commands, can be executed multiple times, consecutively.

2. Loop

A loop is a group of commands in a computer program that are being repeated.

- Each repetition of the loop is called a pass.
- The number of passes can be set to be fixed, or the looping process is terminated when a specified condition is satisfied.
- In each pass some or all of the variables that are defined in the loop obtain new values.
3. for-end Loops
A for–end loop is used when the number of passes is known (or can be calculated) in advance. A variable is used to control the looping process. The general structure of a for–end loop is:

- In the first pass, \( k = f \), and Matlab executes the commands between the for and the end.
- The computer goes back to the for command for the second pass. \( k \) obtains a new value equal to \( k = f + s \), and the commands between the for and the end are executed with the new value of \( k \).
- The process repeats itself until the last pass where \( k = t \).
- **Example:** \( k = 1: 2: 9 \)
  there are five passes through the loop. The values of \( k \) are: 1 3 5 7 9

**Rules Regarding the for Command**

for \( k = f: s: t \)

- The increment value \( s \) can be negative (i.e. \( k = 25: -5: 10 \) produces 4 loops with: \( k = 25, 20, 15, 10 \))
- If \( s \) is omitted, the increment value is 1 (default)
- If \( f \) equals to \( t \), the loop is executed once
- The value of \( k \) should not be redefined within the loop
- The looping continues until the value of \( k \) exceeds the value of \( t \) (i.e. \( k = 8:10:50 \) produces 5 loops with: \( k = 8, 18, 28, 38, 48 \))
### Examples of for–end Loops

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;&gt; for k = 1: 3: 10</td>
<td></td>
</tr>
<tr>
<td>x = k^2</td>
<td></td>
</tr>
<tr>
<td>end</td>
<td></td>
</tr>
<tr>
<td>x =</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>x =</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
</tr>
<tr>
<td>x =</td>
<td></td>
</tr>
<tr>
<td>49</td>
<td></td>
</tr>
<tr>
<td>x =</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td></td>
</tr>
<tr>
<td>&gt;&gt; for k = 5:8</td>
<td></td>
</tr>
<tr>
<td>b = 2*k</td>
<td></td>
</tr>
<tr>
<td>end</td>
<td></td>
</tr>
<tr>
<td>b =</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
</tr>
<tr>
<td>b =</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
</tr>
<tr>
<td>b =</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
</tr>
<tr>
<td>b =</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
</tr>
</tbody>
</table>

If a step value is not entered, the default is 1.

| >> for k = 10:2:20 |                  |
| a = k/3;          |                  |
| end               |                  |
| >> a              |                  |
| a =               |                  |
| 6.6667            |                  |
| >> k              |                  |
| k =               |                  |
| 20                |                  |

A semicolon is used, so `a` is not printed after each pass.

In the last pass: `k = 20`, so then `a = 6.667`
4. for-end Loops in MATLAB

Some basic guidelines for a for-end loop in MATLAB are:

- For every ‘for’ command a computer program **MUST** have an ‘end’ command
- for loops can be used in the command window and in script and function files
- A semicolon is not needed after the ‘for k = m:s:p’ command to suppress printing of k
- To display the value of k in each pass (sometimes useful for debugging) type k as one of the commands in the loop
- Loops can include conditional statements and any other MATLAB commands (functions, plots, etc.)

Example Of Using A for-end Loop In MATLAB

Suppose that you want to use calculated values after the loop is finished – they need to be stored!

Script file:

```matlab
for k=1:4
    b(k)=2*(k+4)
end
```

Command Window:

```
b =  10
b =  10  12
b =  10  12  14
b =  10  12  14  16
```

At the end of the loop, b contains all of the values calculated during the loop. This approach can be used when calculating values which could not be generated by a colon operator and dot operators!
Specific elements can be extracted from an existing array as well:

Script file:

```matlab
a=1:25;
for k=1:5
  ka=5*k;
  b(k)=a(ka)
end
```

Command Window:

```
b = 5
b = 5 10
b = 5 10 15
b = 5 10 15 20
b = 5 10 15 20 25
```

Elements of `a` have been extracted and stored in `b`. This illustrates why the for loop variable is called an index, because this is its most common usage.