Topics: Iteration using for, while
Reading: CFile Chapter 2 (Pay close attention to Sec 2.2 on “floating point” system)

Example 1: Area of $n$-gon

Complete the following program to compute and display the areas of inscribed and circumscribed regular $n$-gons in the unit circle where $n = L, L + 1, \ldots, R - 1, R$. Use only scalar variables.

```matlab
L = input('Enter lower bound for n: ');
R = input('Enter upper bound for n: ');
fprintf('
 n\tInner Area\tOuterArea\n');

% Compute and display areas of n-gons
innerA = (n/2)*sin(2*pi/n);
outerA = n*tan(pi/n);
fprintf('%d\t%.6f\t%.6f\n', n, innerA, outerA);
```

Syntax of the for Loop

```
for <index variable> = <lower bound> : <increment> : <upper bound>
    Statements to execute
    Also called loop body
end
```

The index variable takes on the values specified in the loop header one at a time, i.e., one value for each pass through the loop. The set of values that the index variable will take on is determined at the beginning of the loop. If the set of values is the empty set, then the loop body will not be executed.

Examples of index values:

<table>
<thead>
<tr>
<th>Expression</th>
<th>Set of values that the index variable will take on</th>
</tr>
</thead>
<tbody>
<tr>
<td>3:1:7</td>
<td>3, 4, 5, 6, 7</td>
</tr>
<tr>
<td>2:0.5:3.5</td>
<td>2, 2.5, 3, 3.5 (Non-integer values are OK)</td>
</tr>
<tr>
<td>1:4</td>
<td>1, 2, 3, 4 (Default increment is 1)</td>
</tr>
<tr>
<td>0:-2:-6</td>
<td>0, -2, -5 (“Increment” may be negative)</td>
</tr>
<tr>
<td>5:2:1</td>
<td>∅ (Loop body will not execute)</td>
</tr>
</tbody>
</table>

Iteration

Important features:

- A task can be accomplished if some steps are repeated a number of times; these steps form the loop body
- Must have a starting point
- Must know when to stop
- Must keep track of (and measure) progress

Example 2: “Accumulating” a solution using iteration

Consider the following script. How many passes through the loop will be completed? ____________________
% Average 10 numbers from user input

n= 10;       % number of data values

for k= 1:n
% read and process input value
    num= input('Enter a number: ');
    total= total + num;
end
ave= total/n   % average of n numbers

Example 3: \( n \)-gon \( \rightarrow \) circle

As \( n \) increases, the regular inscribed and circumscribed \( n \)-gons converge to the circle. Since the area of the unit circle is \( \pi \), we have

\[
\lim_{{n \to \infty}} \text{innerArea}_n = \pi \quad \lim_{{n \to \infty}} \text{outerArea}_n = \pi.
\]

Write a program to find \( n \) “sufficiently large” to approximate the area of the unit circle.

Itemize the tasks:

Algorithm:

Syntax of the while Loop

\[
\text{while } \text{condition} \\
\quad \text{statements to execute if expression evaluates to true} \\
\text{end}
\]

If the condition (loop guard) evaluates to true, the loop body executes and the flow of the program goes back to the loop guard—repetition. When the condition evaluates to false, the loop body is skipped and the program continues after the end keyword of the loop.

Two useful patterns

<table>
<thead>
<tr>
<th>Pattern for doing something ( n ) times: definite iteration</th>
<th>Pattern for doing something an indefinite number of times: indefinite iteration</th>
</tr>
</thead>
</table>
| for \( k = 1:n \)                                            | % initialization
| % do something                                               | % ... \textbf{while} not stopping signal
| % ...                                                        | % do something
| end                                                          | % ... % update status (variables)
|                                                             | % ... end |