`for` Loops in MATLAB

Gerald W. Recktenwald
Department of Mechanical Engineering
Portland State University
gerry@pdx.edu

Loops in MATLAB

Repetition or Looping

A sequence of calculations is repeated until *either*

1. All elements in a vector or matrix have been processed

or

2. The calculations have produced a result that meets a predetermined termination criterion

Looping is achieved with `for` loops and `while` loops.

`for` loops

`for` loops are most often used when each element in a vector or matrix is to be processed.

Syntax:

```
for index = expression
    block of statements
end
```

Example: Sum of elements in a vector

```
x = 1:5;            %  create a row vector
sumx = 0;           %  initialize the sum
for k = 1:length(x)
   sumx = sumx + x(k);
end
```

`for` loops

Colon notation is often used in `for` loop constructs, but the colon is not required, and it is not part of the 'for' loop syntax.

In this example

```
x = 1:5;            %  create a row vector
sumx = 0;           %  initialize the sum
for k = 1:length(x)
   sumx = sumx + x(k);
end
```

the `1:length(x)` expression is a row vector from 1 to the number of elements in `x`.

`for` loop variations

Example: A loop with an index incremented by two

```
for k = 1:2:n
   ...
end
```

Example: A loop with an index that counts down

```
for k = n:-1:1
   ...
end
```

`for` loop variations

Example: A loop with non-integer increments

```
for x = 0:pi/15:pi
   fprintf('%8.2f  %8.5f\n',x,sin(x));
end
```

Note: In this example, `x` is a *scalar* inside the loop. Each time through the loop, `x` is set equal to one of the *columns* of `0:pi/15:pi`.

The `fprintf` statement creates formatted output to the command window.