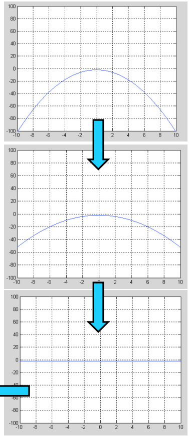
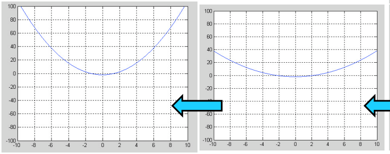
CSCI 130
Introduction to Engineering Computing
Class Meeting #24

Engineering Computing and Problem Solving with Matlab

Matlab graphs with animation
Structured programming in Matlab

Animation

```
clear,clc,clf
x = -10:0.01:10;
k = -1;
y = k*x.^2-2;
h = plot(x,y);
grid on
% set(h,'EraseMode','xor')
axis([ -10 10 -100 100 ])
while k<1
    k=k+0.001;
    y = k*x.^2-2;
    set(h,'Xdata',x,'Ydata',y)
    drawnow
end
```

Structured programming in Matlab

In Matlab code, there are similar programming structures to VBA. VBA's structures are superior, but you can accomplish much of the same with Matlab code.

Operator Precedence [left-to-right, override with ( )]

Arithmetic operators	Relational operators	Logical operators
^	==	& [AND]
unary – (negation)	~= [not equal]	| [OR]
* /	<	~ [NOT]
+ -	>	
	<=	
	>=	

higher precedence ⇐ higher precedence ⇐

Selection structure [decisions]

One-way IF

"If-Then"

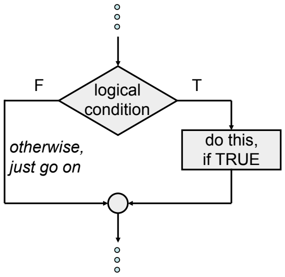
```
if logical expression
   statements
end
```

Example:

```
if Xval > 10
   Yval = Xval – 10;
   Xval = 1;
end
```

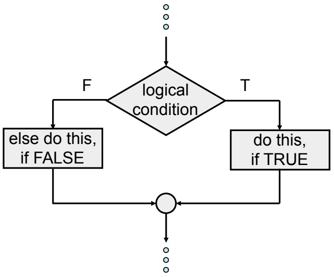
One-line version:

```
if Xval > 10, Yval = Xval – 10;, end
```



Two-way If

"If-Then-Else"



```
if logical expression
   statements
else
  statements
end
```

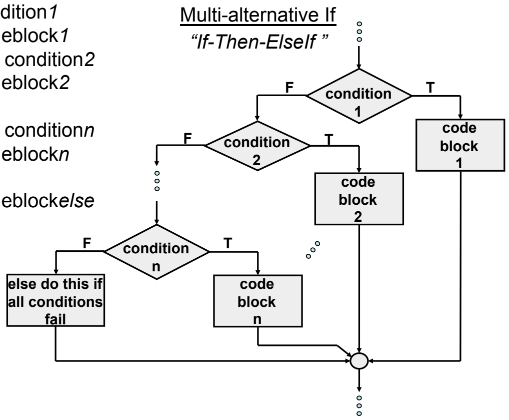
Example:

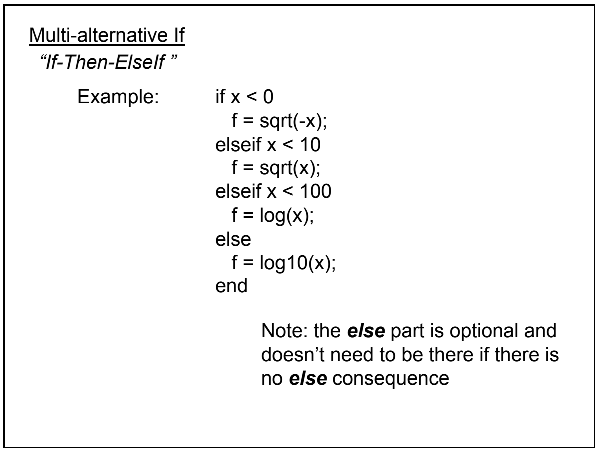
```
if j == 1
   z = sin(x);
else
   nr = 2;
end
```

```
if condition1
   codeblock1
elseif condition2
   codeblock2
  ⋮
elseif conditionn
   codeblockn
else
   codeblockelse
end
```

Multi-alternative If

"If-Then-ElseIf "





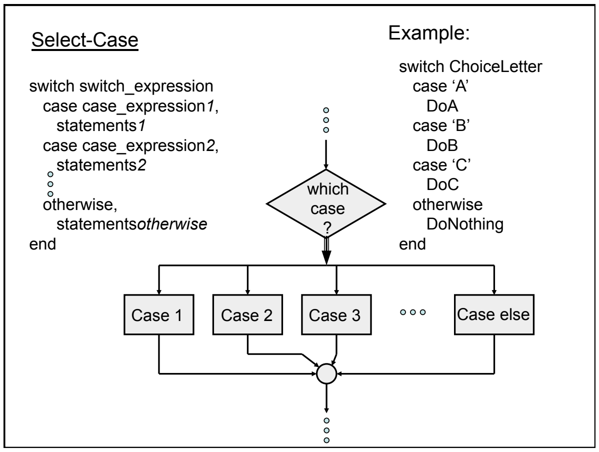
Multi-alternative If

"If-Then-ElseIf "

Example:

```
if x < 0
  f = sqrt(-x);
elseif x < 10
  f = sqrt(x);
elseif x < 100
  f = log(x);
else
  f = log10(x);
end
```

Note: the ***else*** part is optional and doesn't need to be there if there is no ***else*** consequence

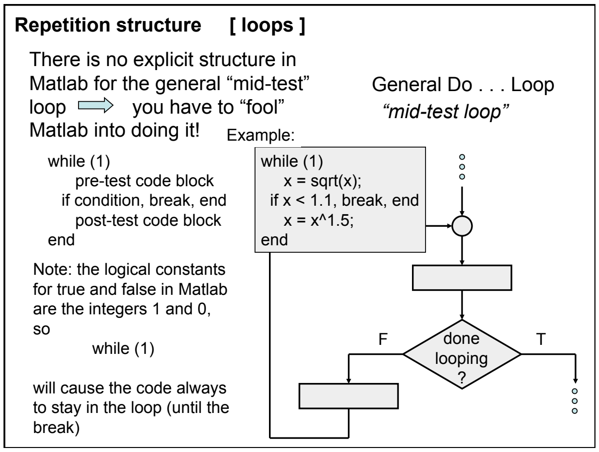


Select-Case

```
switch switch_expression
  case case_expression1,
    statements1
  case case_expression2,
    statements2
  ⁝
  otherwise,
    statementsotherwise
end
```

Example:

```
switch ChoiceLetter
  case 'A'
    DoA
  case 'B'
    DoB
  case 'C'
    DoC
  otherwise
    DoNothing
end
```



Repetition structure [loops]

There is no explicit structure in Matlab for the general "mid-test" loop ⇨ you have to "fool" Matlab into doing it!

```
while (1)
    pre-test code block
  if condition, break, end
    post-test code block
end
```

Example:

```
while (1)
    x = sqrt(x);
  if x < 1.1, break, end
    x = x^1.5;
end
```

General Do . . . Loop

"mid-test loop"

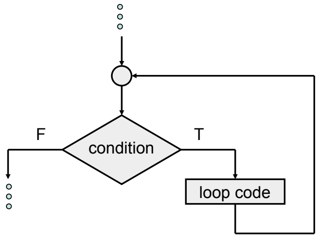
Note: the logical constants for true and false in Matlab are the integers 1 and 0, so

while (1)

will cause the code always to stay in the loop (until the break)

Special Cases of the General Do . . . Loop

Do – While *"pre-test loop"*

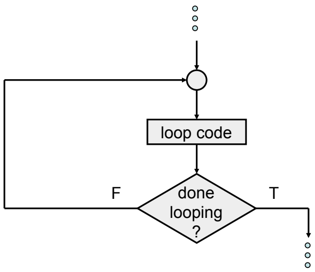


```
while condition
   loop code
end
```

Example:

```
while x > 1.1
   x = sqrt(x);
end
```

Do – Until *"post-test loop"*



again, Matlab must be "fooled" into doing this structure

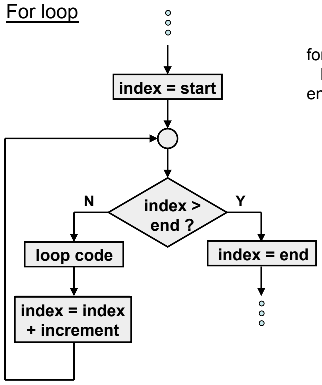
```
while (1)
   loop code
   if condition, break, end
end
```

Example:

```
while (1)
   i = i + 1;
   if i > ilim, break, end
end
```

Count-controlled Iteration

For loop



assumed = +1, if left out

```
for index = start : increment : end
   loop code
end
```

Example:

```
for i = 2 : 7
   x(i) = 2*x(i-1);
end
```

Note: There is a distinction between VBA and Matlab here. In Matlab, the index is at the end value after the loop is exited. In VBA, the index is one step beyond the end value.