#### Boolean algebra, conditional statements, loops.

Notes

Eugeniy E. Mikhailov

The College of William & Mary



Lecture 03

Eugeniy Mikhailov (W&M)	Practical Computing	<□> < <b>∂</b>	> ← ₹ > ← ₹ > — ₹ Lecture 03	୬୧୯ 1/19
Boolean algebra				
Variable of boolean typ • true • false	e can have only two val	lues		

			200
Eugeniy Mikhailov (W&M)	Practical Computing	Lecture 03	2 / 19
Boolean algebra			

Variable of boolean type can have only two values

- true (Matlab use 1 to indicate it, actually everything but zero)
- false

			990
Eugeniy Mikhailov (W&M)	Practical Computing	Lecture 03	2/19
Boolean algebra			

Variable of boolean type can have only two values

- true (Matlab use 1 to indicate it, actually everything but zero)
- false (Matlab uses 0)

-			
Notes			
-			
Notes			
Notes			

(D) (B) (E) (E) E 99

ugeniy Mikhailov (W&M)

Practical Computing

re 03 2

#### Boolean algebra

Variable of boolean type can have only two values

- true (Matlab use 1 to indicate it, actually everything but zero)
- false (Matlab uses 0)

There are three logical operators which are used in boolean algebra

#### Boolean algebra

Variable of boolean type can have only two values

- true (Matlab use 1 to indicate it, actually everything but zero)
- false (Matlab uses 0)

There are three logical operators which are used in boolean algebra

¬ - logic not, Matlab

 $\neg$ true = false

 $\neg$ false = true

#### Boolean algebra

Variable of boolean type can have only two values

- true (Matlab use 1 to indicate it, actually everything but zero)
- false (Matlab uses 0)

There are three logical operators which are used in boolean algebra

• ¬ - logic **not**, Matlab

 $\neg$ true = false

 $\neg$ false = true

• A - logic and, Matlab &

frue, if A=true and B=true, false, otherwise

#### Boolean algebra

Variable of boolean type can have only two values

- true (Matlab use 1 to indicate it, actually everything but zero)
- false (Matlab uses 0)

There are three logical operators which are used in boolean algebra

¬ - logic not, Matlab

¬true = false

 $\neg$ false = true

• A - logic and, Matlab &

frue, if A=true and B=true, false, otherwise

∨ - logic or, Matlab

false, if A=false and B=false, true, otherwise

Notes

Notes

Notes

Boolean operators precedence in Matlab	
If $A = $ false, $B = $ true, $C = $ true	Notes
·	
Eugenly Mikhailov (W&M) Practical Computing Lecture 03 3/19	
Boolean operators precedence in Matlab	Notes
If $A = $ false, $B = $ true, $C = $ true	Notes
$A {\sim}B\&C$	
~ has highest precedence, then &, and then	
→ □ → ← ② → ← ② → ← ② → ② ← ② ← ② ← ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	
Eugeniy Mikhailov (W&M) Practical Computing Lecture 03 3 / 19	
Boolean operators precedence in Matlab	Notes
If $A = $ false, $B = $ true, $C = $ true	
$A {\sim}B\&C$	
$\sim$ has highest precedence, then $\&$ , and then	
$A   ((\sim B) \& C)$	
4□> ⟨₫> ⟨ξ> ⟨ξ⟩ ½ √Q.	
Eugenly Mikhailov (W&M) Practical Computing Lecture 03 3 / 19	
Boolean operators precedence in Matlab	Notes
If $A = false$ , $B = true$ , $C = true$	
$A {\sim}B\&C$	
$\sim$ has highest precedence, then &, and then $\mid$	
$A   ((\sim B) \& C)$	-
Thus	

 $A|{\sim}B\&C=\mathit{false}$ 

#### Boolean operators precedence in Matlab

If A = false, B = true, C = true

 $A|{\sim}B\&C$ 

 $\sim$  has highest precedence, then &, and then

Notes
Notes
Notes
-

#### Boolean logic examples

There is an island, which is populated by two kind of people: liars and truthlovers.

Notes

- Liars always lie and never speak a word of truth.
- Truthlovers always speak only truth.

Suppose, you are landed on this island and met a person. What will be the answer to your question "Who are you?"

Eugeniy Mikhailov (W&M)

Practical Computing

Lecture 03 4/19

Boolean logic examples

There is an island, which is populated by two kind of people: liars and truthlovers.

- Liars always lie and never speak a word of truth.
- Truthlovers always speak only truth.

Suppose, you are landed on this island and met a person. What will be the answer to your question "Who are you?"

• The answer always will be "Truthlover".

Eugeniy Mikhailov (W&M)

Practical Computing

Lecture 03 4/19

Boolean logic examples

There is an island, which is populated by two kind of people: liars and truthlovers.

- Liars always lie and never speak a word of truth.
- Truthlovers always speak only truth.

Suppose, you are landed on this island and met a person. What will be the answer to your question "Who are you?"

• The answer always will be "Truthlover".

Now you see a person who answers to your question. "I am a liar." Is it possible?

Eugenly Mikhailov (W&M)

Practical Computing

Lecture 03 4 / 19

Boolean logic examples

There is an island, which is populated by two kind of people: liars and truthlovers.

- Liars always lie and never speak a word of truth.
- Truthlovers always speak only truth.

Suppose, you are landed on this island and met a person. What will be the answer to your question "Who are you?"

• The answer always will be "Truthlover".

Now you see a person who answers to your question. "I am a liar." Is it possible?

• This makes a paradox and should not ever happen on this island.

-			
Notes			
-			
Notes			
-			
Notos			
Notes			

Matlab boolean logic examples	
● 123.3 & 12=	Notes
Eugenly Mikhailov (W&M)  Practical Computing  Lecture 03 5/19  Matlab boolean logic examples	
• 123.3 & 12=1 • ~ 1232e-6 =	Notes
Eugenly Mikhailov (W&M)  Practical Computing  Lecture 03 5/19  Matlab boolean logic examples	Notes
<ul> <li>123.3 &amp; 12= 1</li> <li> ~ 1232e-6 = 0</li> </ul>	Notes
Eugeniy Mikhailov (W&M)  Practical Computing  Lecture 03 5/19  Matlab boolean logic examples	
<ul> <li>123.3 &amp; 12=1</li> <li> ~ 1232e-6 = 0</li> </ul>	Notes
>> B=[1.22312, 0; 34.343, 12] B =	
1.2231 0 34.3430 12.0000	

(D) (B) (E) (E) E 90

Eugeniy Mikhailov (W&M)

Practical Computing

ng

Lecture 03 5 / 19

## Matlab boolean logic examples Notes • 123.3 & 12=**1** • $\sim 1232e-6 = 0$ >> B=[1.22312, 0; 34.343, 12] 1.2231 0 34.3430 12.0000 ~B Lecture 03 5 / 19 Matlab boolean logic examples Notes • 123.3 & 12=**1** • $\sim$ 1232e-6 = **0** >> B=[1.22312, 0; 34.343, 12] 1.2231 0 34.3430 12.0000 ~B 0 1 0 0 Lecture 03 Eugeniy Mikhailov (W&M) Matlab boolean logic examples Notes • 123.3 & 12=**1** • $\sim 1232e-6 = 0$ >> B=[1.22312, 0; 34.343, 12] В = 1.2231 34.3430 12.0000 ~B ans = 0 1 0 0 Matlab boolean logic examples Notes • 123.3 & 12=**1** • $\sim 1232e-6 = 0$ >> B=[1.22312, 0; 34.343, 12] В = 1.2231 0 34.3430 12.0000 ~B ans = 0 1 B | ~B ans =

"To be or not to be"

The answer is to be Eugeniy Mikhailov (W&M)

1

#### Matlab boolean logic examples

#### 

Lecture 03 6 / 19

## Matlab boolean logic examples

Eugeniy Mikhailov (W&M)

0 24.4000

#### B&A

## Eugeniy Mikhailov (W&M) Practical Computing Matlab boolean logic examples

#### 

## Eugeniy Mikhailov (W&M) Practical Computing Matlab boolean logic examples

B&A		A   ~B
ans =		
1	0	
0	1	

《ロ・・タ・・ミ・ミ・ラ・ペーン・ (W&M) Practical Computing Lecture 03 6 / 15

Notes			
Notes			
Notes			
Notes			
Notes			

Notes			

#### Matlab boolean logic examples

B&A	A   ~B		
ans =	ans =	=	
1 0	1	1	
0 1	0	1	

Comparison operators

Math	Matlab
=	== double equal sign!
$\neq$	~=
<	<
$\leq$	<=
>	>
$\geq$	>=

# Eugeniy Mikhailov (W&M) Practical Computing Lecture 03 7/19 Comparison operators

Math	Matlab
=	== double equal sign!
$\neq$	~=
<	<
$\leq$	<=
>	>
>	>=

x=[1,2,3,4,5] x = 1 2 3 4 5

# Eugeniy Mikhaliov (W&M) Practical Computing Comparison operators

#### 

x >= 3

Notes			
Notes			
Notes			
Notes			

#### Comparison operators

Math	Matlab
=	== double equal sign!
<b>≠</b>	~=
<	<
$\leq$	<=
>	>
>	>=

Notes

```
x = [1, 2, 3, 4, 5]

x = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ x & > & 3 & 4 & 5 \end{bmatrix}
```

ans =				
0	0	1	1	1

		4 m > 4 m >
Eugeniy Mikhailov (W&M)	Practical Computing	Lecture 03 7 / 19
Comparison opera	ators	

# Math Matlab = == double equal sign! ≠ ~= <</td> <</td> ≤ <=</td> > > > >

```
x=[1,2,3,4,5]
x =
    1    2    3    4    5

x >= 3

ans =
    0    0    1    1    1
% chose such 'x' where x>=3
    x (x >= 3)
```

		40+40+42+42+3	200
Eugeniy Mikhailov (W&M)	Practical Computing	Lecture 03	7 / 19
Comparison opera			

Math	Matlab
=	== double equal sign!
$\neq$	~=
<	<
$\leq$	<=
>	>
$\geq$	>=

#### Comparison with matrices

Notes			
N .			
Notes			

#### Comparison with matrices

>>	A=[1,2;3,4]	
A =	:	
1	2	
2	1	

	4		Þ	4 10	ŀ	4	E	Þ	4	E	Þ	8	200	
--	---	--	---	------	---	---	---	---	---	---	---	---	-----	--

### Eugeniy Mikhailov (W&M)

Practical Computing

Lecture 03 8 / 19

#### Comparison with matrices

>>	A=[1,2;3,4]
A =	-
1	2
3	4

A>=2

Eugeniy Mikhailov (W&M)

### ← □ ト ← □ ト ← 필 ト ← 필 → ○ ○ ○ Lecture 03 8 / 19

#### Comparison with matrices

A>=2

A(A>=2)

ans = 0 1

#### (마) (레) (크) (크) (크) 연(C)

#### Comparison with matrices

A>=2

ans	=			
0		1		
1		1		

#### Notes

#### Notes

#### Notes

·	•	•	•

#### Comparison with matrices

>>	A=[1,2;3,4]	
A =	:	
1	2	
_		

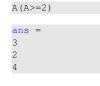
#### A>=2 ans = 0 1

Chose such
elements of B where
elements of A≥2

Eugeniy Mikhail	lov (W&M)	

#### Comparison with matrices

A>=2 0 1 1



B(A>=2) Chose such elements of B where elements of A>2

ans = 53 11 42

#### if-else-end statement

if expression this part is executed only if expression is true else this part is executed

only if expression is false

end

#### if-else-end statement

if expression this part is executed only if expression is true

if hungry buy some food else this part is executed keep working

only if *expression* is false end

Ν	lotes

-		

#### Notes

#### Notes

#### if-else-end statement

if expression this part is executed only if expression is true

else

if hungry buy some food

this part is executed keep working

only if expression is false end

if (x>=0)y=sqrt(x);else error('cannot do');

#### Common mistake in the 'if' statement

```
if (x=y)
  D=4;
  Z = 45;
 C=12;
else
 D=2;
end
```

Lecture 03

#### Common mistake in the 'if' statement

```
if (x=y)
 D=4;
 Z = 45;
 C=12;
else
 D=2;
```

the value of 'D' is always 4, except the case when y=0

#### Common mistake in the 'if' statement

```
if (x=y)
 D=4;
  Z = 45;
 C=12;
else
 D=2;
end
```

the value of 'D' is always 4, except the case when y=0someone used assignment operator (=) instead of comparison (==) Notes

Notes

Notes

# Short form of 'if-end' statement if expression this part is executed only if *expression* is true end

		4	□ > < <b>∅</b> > < 3	< ± > < ± >	2	900
Eugeniy Mikhailov (W&M)	Practical Computing			Lecture 03		11 / 19
Short form of 'if-ei	nd' statement					

if expression this part is executed if won a million only if expression is go party end true end

		<ul><li>(ロ) (問) (注) (注) 注 り(0</li></ul>	
Eugeniy Mikhailov (W&M)	Practical Computing	Lecture 03 11 / 19	ı
Short form of 'if-e	nd' statement		

if expression this part is executed if won a million only if expression is go party exit; end end true end

if (deviation<=0)</pre>

		< D > < B	) 	990
Eugeniy Mikhailov (W&M)	Practical Computing		Lecture 03	11 / 19
The 'while' statement				
while expression				
this part is executed while <i>expression</i> is				
true				

Notes			
Notes			
Notes			

Notes

Eugeniy Mikhailov (W&M)

end

#### The 'while' statement

while expression this part is executed while expression is

while hungry keep eating

end

true end

Lecture 03 The 'while' statement

while expression this part is executed while expression is

end

true

end

end

keep eating

end

while (i<=10) while hungry c=a+b;z = c \* 4 + 5;i=i+2;end

i=1;

The 'while' statement i=1; while expression while (i<=10) this part is executed while hungry c=a+b; while expression is keep eating

z=c\*4+5;

i=i+2;

end

while loop is extremely useful but they are not guaranteed to finish. For a bit more complicated conditional statement and loop it is impossible to predict if the loop will finish.

The 'while' statement

while expression while hungry this part is executed while expression is keep eating true end

end

z=c\*4+5;i=i+2;

i=1;

while (i<=10)

c=a+b;

while loop is extremely useful but they are not guaranteed to finish. For a bit more complicated conditional statement and loop it is impossible to predict if the loop will finish.

Yet another common mistake is

i=1; while (i<=10) c=a+b;end

Notes	
Mater	
Notes	
Notes	
Notes	

Notes			
-			

#### The 'while' statement

while expression
this part is executed while hungry
while expression is true end

```
i=1;
while (i<=10)
    c=a+b;
    z=c*4+5;
    i=i+2;
end</pre>
```

while loop is extremely useful but they are not guaranteed to finish. For a bit more complicated conditional statement and loop it is impossible to predict if the loop will finish.

Yet another common mistake is

```
i=1;
while (i<=10)
    c=a+b;
end</pre>
```

not updating the term leading to fulfillment of the while condition

Eugeniy Mikhailov (W&M)

Practical Computing

Lecture 03

#### The 'for' statement

for variable = expression
do something
end

In this case variable is assigned concequently with columns of the *expression*, and then statements inside of the loop are executed

# Eugenly Mikhailov (W&M) Practical Computing Lecture 03 13/19 The 'for' statement

for variable = expression
do something
end

In this case variable is assigned concequently with columns of the expression, and then statements inside of the loop are executed sum =

```
sum=0;
x=[1,3,5,6]
for v=x
   sum=sum+v;
end
```

>> sum sum = 15

# Eugenly Mikhallov (W&M) Practical Computing Lecture 03 13/1 The 'for' statement

for variable = expression do something

end

In this case variable is assigned concequently with columns of the *expression*, and then statements inside of the loop are executed

```
sum=0;
x=[1,3,5,6]
for v=x
   sum=sum+v;
end
```

>> sum sum = 15

for loops are guaranteed to complete after predictable number of iterations (the amount of columns in *expression*).

Notes		
Notes		
Notes		
Notes		
	<u> </u>	 

#### Example

$$S = \sum_{i=1}^{100} i = 1 + 2 + 3 + 4 + \dots + 99 + 100$$

Lecture 03 14 / 19

#### Example

$$S = \sum_{i=1}^{100} i = 1 + 2 + 3 + 4 + \dots + 99 + 100$$

S=0; i=1; while (i<=100) S=S+i; i=i+1;end

Lecture 03

## Example

$$S = \sum_{i=1}^{100} i = 1 + 2 + 3 + 4 + \dots + 99 + 100$$

S=0; i=1;while (i<=100) S=S+i; i=i+1; end

S=0; for i=1:100 S=S+i;end

Example

While k<=100 and  $a_k \ge 10^{-5}$ , where  $a_k = k^{-k}$ .

Notes

Notes

Notes

#### Example

$$S = \sum_{k=1}^{\infty} a_k$$

While k<=100 and  $a_k \ge 10^{-5}$ , where  $a_k = k^{-k}$ .

```
S=0; k=1;
while ((k \le 100) \& (k^-k \ge 1e-5))
 S=S+k^-k;
 k=k+1;
end
```

## Example

Lecture 03 15 / 19

$$S = \sum_{k=1}^{\infty} a_k$$

While k<=100 and  $a_k \ge 10^{-5}$ , where  $a_k = k^{-k}$ .

```
S=0; k=1;
while ( (k \le 100) \& (k^-k \ge 1e-5) )
 S=S+k^-k;
 k=k+1;
end
```

#### Lecture 03

## Example

While k<=100 and  $a_k \ge 10^{-5}$ , where  $a_k = k^{-k}$ .

```
S=0; k=1;
                                    S=0; k=1;
while ( (k \le 100) & (k^-k \ge 1e-5) )
                                    while( k<=100 )
 S=S+k^-k;
                                      a_k=k^-k;
 k=k+1;
                                      if (a_k < 1e-5)
end
                                       break;
                                      end
>> S
                                      S=S+a_k;
S =
                                      k=k+1;
1.2913
                                    end
```

#### Example

$$S = \sum_{k=1}^{n} a_k$$

While k<=100 and  $a_k \ge 10^{-5}$ , where  $a_k = k^{-k}$ .

>> S	
S =	
1.2913	

_	_	_
.ecture	03	15 /

Notes			

Notes

Notes

#### Same example with 'for' loop and use of matrix ops

$$S = \sum_{k=1} a_k$$

While k<=100 and  $a_k \ge 10^{-5}$ , where  $a_k = k^{-k}$ .

4 D > 4 B > 4 E > 4 E > E + 9 Q C

Eugeniy Mikhailov (W&M)

Practical Computing

ecture 03 16 / 19

Notes

#### Same example with 'for' loop and use of matrix ops

$$S = \sum_{k=1} a_k$$

While k<=100 and  $a_k \ge 10^{-5}$ , where  $a_k = k^{-k}$ .

```
S=0;
for k=1:100
   a_k=k^-k;
   if (a_k < 1e-5)
        break;
   end
   S=S+a_k;
end</pre>
```

Eugeniy Mikhailov (W&M)

Practical Computing

ıre 03 16 / 19

#### Same example with 'for' loop and use of matrix ops

$$S = \sum_{k=1} a_k$$

While k<=100 and  $a_k \ge 10^{-5}$ , where  $a_k = k^{-k}$ .

S=0;
for k=1:100
 a\_k=k^-k;
 if (a\_k < 1e-5)
 break;
 end
 S=S+a\_k;
end</pre>

Often it is more elegant to use built in Matlab matrix operators

#### Note

- >> S S =
- use of the *choose* elements construct
- built in sum function

Eugeniy Mikhailov (W&M)

Practical Computing

(2) (2) 2

#### Interest rate related example

Suppose bank gave you 50% interest rate (let's call it 'x'), and you put one dollar in.

How much would you get at the end of the year?

one payment at the end of the year

$$M_1 = 1 * (1 + x) = 1 * (1 + .5) = 1.5$$

	_
	-
	_
	-
	-
	-
	-
Notes	
	-
	-
	-
	-
	-
	-
Notes	
	-
	-
	-
	_
Notes	
	-
	_
	-
	_
	-

D > 4 @ > 4 E > 4 E > E 994

Eugeniy Mikhailov (W&M)

Practical Computing

Lecture 03

#### Interest rate related example

Suppose bank gave you 50% interest rate (let's call it 'x'), and you put one dollar in.

Notes

How much would you get at the end of the year?

one payment at the end of the year

$$M_1 = 1 * (1 + x) = 1 * (1 + .5) = 1.5$$

interest payment every half a year

$$M_2 = 1 * (1 + x/2) * (1 + x/2) = 1 * (1 + .5/2)^2 = 1.5625$$

#### Interest rate related example

Suppose bank gave you 50% interest rate (let's call it 'x'), and you put

How much would you get at the end of the year?

one payment at the end of the year

$$M_1 = 1 * (1 + x) = 1 * (1 + .5) = 1.5$$

interest payment every half a year

$$M_2 = 1 * (1 + x/2) * (1 + x/2) = 1 * (1 + .5/2)^2 = 1.5625$$

• interest payment every month

$$M_{12} = 1 * (1 + x/12)^{12} = 1.6321$$

Lecture 03

#### Interest rate related example

Now let's find how your return on investment  $(M_N)$  depends on the number of payments per year

x=.5; N\_max=100; N=1:N\_max; M=0\*(N); % since N is vector M will be a vector too for i=N  $M(i) = (1+x/i)^i;$ end plot(N,M,'-'); set(gca,'FontSize',24); xlabel('N, number of payments per year'); ylabel('M\_n, return on investment'); % note M\_n use title('Return on investment vs number of payments');

Of course we do not need computer to show that  $M_{\infty}=e^{x}=1.6487$ but we need it to calculate something like

 $M_{1001} - M_{1000} = 2.0572 \times 10^{-7}$ 

#### Interest rate related example

Now let's find how your return on investment  $(M_N)$  depends on the number of payments per year

x=.5; N\_max=100; N=1:N\_max; M=0\*(N); % since N is vector M will be a vector too for i=N  $M(i) = (1+x/i)^i;$ plot(N,M,'-'); set(gca,'FontSize',24); xlabel('N, number of payments per year'); yl ti

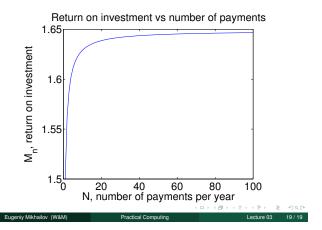
M

Вс

	rn on investment') investment vs numb		;			
course we do not not the course we do not not the contract we need it to calculate $M_{1001} - M_{1000} = 2.057$		at $M_{\infty}=e^x=1.6487$				
onus question: can you calculate M without use of loops?						
nus question: can y	ou calculate M without use	e of loops?	J			
onus question: can ye Eugeniy Mikhailov (W&M)	ou calculate M without use  Practical Computing	·	/ 19			
		·	/ 19			

Notes			
Notes			
Notes			
Notes			

#### Interest rate related example



Notes	
	_
	_
	_
	_
	_
	_
Notes	
Notes	
	_
	_
	_
	_
	_
	_
	_
Notes	
	_
	_