### Boolean algebra, conditional statements, loops.

Eugeniy E. Mikhailov

The College of William & Mary



Lecture 03

		40+40+42+42+ 2	200
Eugeniy Mikhailov (W&M)	Practical Computing	Lecture 03	
Boolean algebra			

Variable of boolean type can have only two values

- true
- false

		40>40>45>45>	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

		< + + >	(E) (E) (E)	200
Eugeniy Mikhailov (W&M)	Practical Computing		Lecture 03	2 / 19
Boolean algebra				
Variable of backers turn	a aan haya anly twa	(aluea		

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		
140103		
-		
Notes		
Notes		

### Boolean algebra

Variable of boolean type can have only two values

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

Notes

Notes

Notes

• false (Matlab uses 0)

There are three logical operators which are used in boolean algebra

		+ = > + <del>   </del> > + \  \  > >	4 € > €	990
Eugeniy Mikhailov (W&M)	Practical Computing	Į.	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)

There are three logical operators which are used in boolean algebra

• ¬ - logic **not**, Matlab ^

 $\neg true = false$ 

¬false = true

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)

There are three logical operators which are used in boolean algebra

• ¬ - logic **not**, Matlab

 $\neg$ true = false  $\neg$ false = true

•  $\wedge$  - logic **and**, Matlab &

$$A \wedge B = \begin{cases} \text{true, if A=true and B=true,} \\ \text{false, otherwise} \end{cases}$$

Cugeniy Mikhailov (W&M)

Cugeniy Mikhailov (W&M)

Practical Computing

Cugeniy Mikhailov (W&M)

### 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 &

 $A \wedge B = \begin{cases} \text{true, if A=true and B=true,} \\ \text{false, otherwise} \end{cases}$ 

● ∨ - logic **or**, Matlab

$$A \lor B = \begin{cases} false, & \text{if A=false and B=false}, \\ frue, & \text{otherwise} \end{cases}$$

niy Mikhailov (W&M) Practical Computing Lecture 03 2

Notes			

Boolean operators precedence in Matlab	
If $A = $ false, $B = $ true, $C = $ true	Notes
A ∼B& <i>C</i>	
'	
《ロシャグラ・《言》を言う を ず)な(? 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	
(D) (B) (E) (E) E OQ	
Eugeniy Mikhailov (W&M) Practical Computing Lecture 03 3 / 19	
Boolean operators precedence in Matlab	Notes
Boolean operators precedence in Matlab  If $A = false, B = true, C = true$	Notes
Boolean operators precedence in Matlab	Notes
Boolean operators precedence in Matlab  If $A = false$ , $B = true$ , $C = true$ $A {\sim}B\&C$ $\sim \text{ has highest precedence, then \&, and then } $	Notes
Boolean operators precedence in Matlab	Notes
Boolean operators precedence in Matlab  If $A = false$ , $B = true$ , $C = true$ $A {\sim}B\&C$ $\sim \text{ has highest precedence, then \&, and then } $	Notes
Boolean operators precedence in Matlab  If $A = false$ , $B = true$ , $C = true$ $A {\sim}B\&C$ $\sim \text{ has highest precedence, then \&, and then } $	Notes
Boolean operators precedence in Matlab  If $A = false$ , $B = true$ , $C = true$ $A {\sim}B\&C$ $\sim \text{ has highest precedence, then \&, and then } $	Notes
Boolean operators precedence in Matlab  If $A = false$ , $B = true$ , $C = true$ $A {\sim}B\&C$ $\sim \text{ has highest precedence, then \&, and then } $	Notes
Boolean operators precedence in Matlab  If $A = false$ , $B = true$ , $C = true$ $A {\sim}B\&C$ $ ^{\bullet}$ has highest precedence, then &, and then   $A \left(({\sim}B)\&C\right)$	Notes
Boolean operators precedence in Matlab  If $A = false$ , $B = true$ , $C = true$ $A {\sim}B\&C$ $ ^{\bullet}$ has highest precedence, then &, and then   $A \left(({\sim}B)\&C\right)$	Notes
Boolean operators precedence in Matlab  If $A = false$ , $B = true$ , $C = true$ $A {\sim}B\&C$ $ ^{\bullet}$ has highest precedence, then &, and then   $A \left(({\sim}B)\&C\right)$	Notes
Boolean operators precedence in Matlab  If $A = false$ , $B = true$ , $C = true$ $A {\sim}B\&C$ $ ^{\bullet}$ has highest precedence, then &, and then   $A \left(({\sim}B)\&C\right)$	
Boolean operators precedence in Matlab  If $A = false$ , $B = true$ , $C = true$ $A \sim B\&C$ $\sim \text{ has highest precedence, then }\&, \text{ and then } $ $A ((\sim B)\&C)$ Eugeniy Mikhailov (W&M) Practical Computing Lecture 03 3/19  Boolean operators precedence in Matlab	
Boolean operators precedence in Matlab  If $A = false$ , $B = true$ , $C = true$ $A {\sim}B\&C$ $ {\sim} \text{ has highest precedence, then \&, and then }   A (({\sim}B)\&C)$ Eugeniy Mikhailov (W&M) Practical Computing Lecture 03 3/19  Boolean operators precedence in Matlab	
Boolean operators precedence in Matlab  If $A = false$ , $B = true$ , $C = true$ $A {\sim}B\&C$ $\sim \text{ has highest precedence, then \&, and then } $ $A ((\sim B)\&C)$ Eugenly Mikhailov (W&M) Practical Computing Lecture 03 3/19  Boolean operators precedence in Matlab  If $A = false$ , $B = true$ , $C = true$ $A {\sim}B\&C$	
Boolean operators precedence in Matlab  If $A = false$ , $B = true$ , $C = true$ $A \sim B\&C$ $\sim \text{ has highest precedence, then \&, and then } $ $A  ((\sim B)\&C)$ Eugenly Mikhailov (W&M)  Practical Computing  Eugenly Mikhailov (W&M)  A  ((\sim B)\&C)  Thus	
Boolean operators precedence in Matlab  If $A = false$ , $B = true$ , $C = true$ $A \sim B\&C$ $\sim \text{ has highest precedence, then \&, and then } $ $A ((\sim B)\&C)$ Eugenly Mikhallov (W&M) Practical Computing  Boolean operators precedence in Matlab  If $A = false$ , $B = true$ , $C = true$ $A \sim B\&C$ $\sim \text{ has highest precedence, then \&, and then } $ $A ((\sim B)\&C)$	

## Boolean operators precedence in Matlab

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

 $A|\sim B\&C$ 

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

 $A|\left(({\sim}B)\&C\right)$ 

Thus

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

"Cat is an animal and cat is not an animal"

Notes

Boolean operators precedence in Matlab

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

 $A \sim B\&C$ 

~ has highest precedence, then &, and then |

 $A|((\sim B)\&C)$ 

Thus

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

"Cat is an animal and cat is not an animal" is false statement

### Boolean operators precedence in Matlab

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

 $A|\sim B\&C$ 

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

 $A|((\sim B)\&C)$ 

Thus

 $A|\sim B\&C = false$ 

"Cat is an animal and cat is not an animal" is false statement

 $\sim$  Z&Z=

### Boolean operators precedence in Matlab

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

 $A|\sim B\&C$ 

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

 $A|((\sim B)\&C)$ 

Thus

 $A|\sim B\&C = false$ 

"Cat is an animal and cat is not an animal" is false statement

 $\sim$  Z&Z= false

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?"

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".

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?

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?

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

Notes	
Notes	
Notes	

Matlab boolean logic examples	
● 123.3 & 12=	Notes
・ロン・(型)・(差)・(差)・ 差 ぐ) Q (で Eugeniy Mikhailov(W&M) Practical Computing Lecture 03 5/19	
Matlab boolean logic examples	
● 123.3 & 12= <b>1</b> ● ~ 1232e-6 =	Notes
• 10 1252e 0 =	
・ロ・・酉・・き・・き・少くで	
Eugeniy Mikhailov (W&M) Practical Computing Lecture 03 5/19  Matlab boolean logic examples	
• 123.3 & 12= <b>1</b>	Notes
• $\sim 1232e-6 = 0$	
(ロ) (質) (差) (差) (差) (を) かくで	
Eugeniy Mikhailov (W&M) Practical Computing Lecture 03 5/19 Matlab boolean logic examples	
• 123.3 & 12=1	Notes
• $\sim 1232e-6 = 0$	
>> B=[1.22312, 0; 34.343, 12] B =	
1.2231 0 34.3430 12.0000	

```
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
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"
```

Eugeniy Mikhailov (W&M)

### Matlab boolean logic examples

```
>> B=[1.22312, 0; 34.343, 12]
в =
1.2231 0
34.3430 12.0000
>> A=[56, 655; 0, 24.4]
56.0000 655.0000
0 24.4000
```

## Lecture 03 6 / 19

### Matlab boolean logic examples

```
>> B=[1.22312, 0; 34.343, 12]
в =
1.2231 0
34.3430 12.0000
>> A=[56, 655; 0, 24.4]
A =
56.0000 655.0000
0 24.4000
```

### B&A

#### Eugeniy Mikhailov (W&M) Lecture 03 6 / 19 Practical Computing Matlab boolean logic examples

```
>> B=[1.22312, 0; 34.343, 12]
в =
1.2231 0
34.3430 12.0000
>> A=[56, 655; 0, 24.4]
A =
56.0000 655.0000
0 24.4000
```

### B&A ans = 1 0 0

## Eugeniy Mikhailov (W&M)

### Matlab boolean logic examples

```
>> B=[1.22312, 0; 34.343, 12]
B =
1.2231
34.3430 12.0000
>> A=[56, 655; 0, 24.4]
56.0000 655.0000
0 24.4000
```

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

Notes			
Votes			
Votes			
Notos			
Notes			

### Matlab boolean logic examples

B&A	A   ~B
ans =	ans =
1 0	1 1
0 1	0 1
	4 m > 4 m >

### Comparison operators

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

# 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 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
```

x >= 3

-				

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}

x > = 3
```

Eugeniy	Mikhailo	v (W&M)			Practical Computing	Lecture 03
0	0	1	1	1		

### Comparison operators

ans =

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

```
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)
```

# Eugenly Mikhailov (W&M) Practical Computing Lecture 03 7 / 18 Comparison operators

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

### Comparison with matrices

>> A=[1,2;3,4]	>> B=[33,11;53,42]			
A =	В =			
1 2	22 11			
3 4	53 42			

Notes			
140163			
Notes			
Notes			

### Comparison with matrices

>>	A=[1,2;3,4]
A =	•
1	2
3	4

Luguin	· · · · · · · · · · · · · · · · · · ·	(*******)	

### Comparison with matrices

### A>=2

### Lecture 03

## Comparison with matrices

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

### A>=2

A(A>=2)

ans = 0 1

### Comparison with matrices

>> B=[33,11;53,42] В = 22 11 53 42

A>=2 ans =

0 1 1 1

A(A>=2)ans = 3

#### Notes

#### Notes

### Notes

### Comparison with matrices

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

### A>=2 ans = 0 1

```
A(A>=2)
ans =
3
2
```

11	L		
42	2		

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



### 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

Eugeniy Mikhailov (W&M)

#### Notes

### Notes

### if-else-end statement

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

false end

if hungry buy some food

this part is executed keep working

only if expression is

Notes

if (x>=0)

else

y=sqrt(x);

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

## Short form of 'if-end' statement Notes if expression this part is executed only if expression is true end Short form of 'if-end' statement Notes if expression this part is executed if won a million only if expression is go party true end end Short form of 'if-end' statement Notes $\mathtt{if}\ expression$ if won a million if (deviation<=0)</pre> this part is executed only if expression is go party exit; end end true end The 'while' statement Notes while expression this part is executed while expression is true end

### The 'while' statement

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

end

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;end i=i+2;end end

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;true end i=i+2;end 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 i=1; while expression while (i<=10) while hungry this part is executed c=a+b;while expression is keep eating z=c\*4+5;true end 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.

Yet ano

```
i=1;
while
 c=a
end
```

et another common mistake is	
=1; hile (i<=10) c=a+b; nd	
< □ > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē > < Ē	990
Eugeniy Mikhailov (W&M) Practical Computing Lecture 03	12 / 19

Notes
Notes
NOTES
Notes
Notes

### The 'while' statement

end

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 1

Notes

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

Eugeniy Mikhallov (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

Eugeniy Mikhailov (W&M) Practical Computing Lecture 03 13/15

### 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			

### Example

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

(D) (B) (E) (E) (O)

Lugorily Milatanov (vvain)

Practical Computing

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</pre>

4 D > 4 B > 4 E > 4 E > 9 Q Q

Eugeniy Mikhailov (W&M)

Practical Computing

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</pre>

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

+ □ > + @ > + 원 > + 원 > 원 + 의 Q @ ...

### Example

$$S=\sum_{i=1}^{n}a_{i}$$

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

Notes

Notes

Notes

### Example

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 15 / 19

### 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;
```

### Lecture 03 15 / 19

### 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

$$S = \sum_{i=1}^{n} a_i$$

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

>> S 1.2913

Notes
-------

#### Notes

### Notes

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

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

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

(D) (B) (E) (E) E 900

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>
```

4 D > 4 D > 4 E > 4 E > E 900

Eugeniy Mikhailov (W&M)

Practical Computing

ecture 03 16 / 19

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

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

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

121121 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	
	_
	—
	_

### 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

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

Eugeniy Mikhailov(W&M) Practical Computing Lecture 03 17/19

### 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

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

• 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

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

Eugeniy Mikhailov (W&M)

Practical Computing

Lecture 03 17/19

Interest rate related example

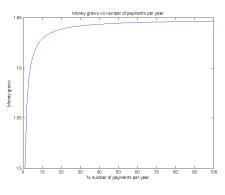
Now let's find how you money growth  $(M_N)$  depends on the number of payments per year

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,'-');
xlabel('N, number of payments per year');
ylabel('Money grows');
title('Money grows vs number of payments per year');

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

 $\begin{array}{c} \textit{M}_{1001} - \textit{M}_{1000} = 2.0572 \times 10^{-7} \\ \\ \textit{Eugenly Mikhailov (W&M)} \end{array} \\ \begin{array}{c} \textit{Practical Computing} \end{array} \\ \begin{array}{c} \textit{Lecture 03} \end{array} \\ \begin{array}{c} \textit{18/19} \end{array}$ 

### Interest rate related example



4	Þ	4	0	Þ	4	2	Þ	4	3	Þ	3	9	٩	C

eniy Mikhailov (W&M) Practical Computing

Notes				
Notes				
	Notes			
Notes	Notes			
Votes				
Notes				
	Notes			